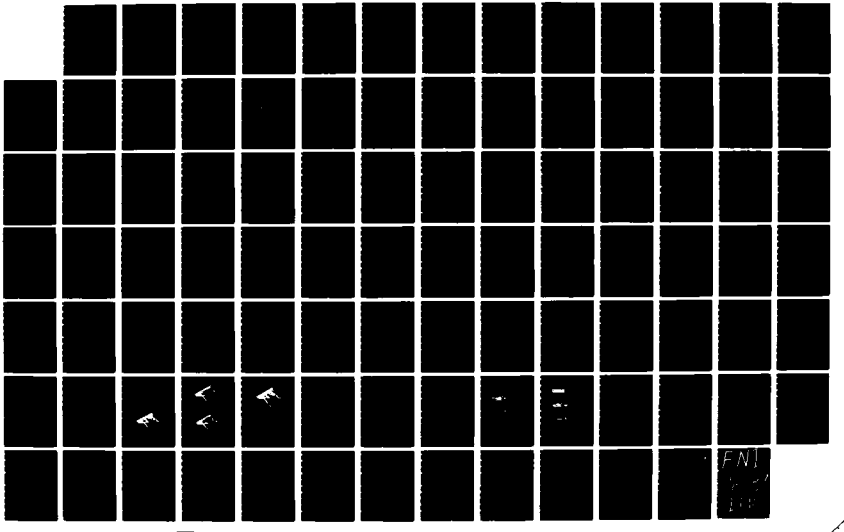
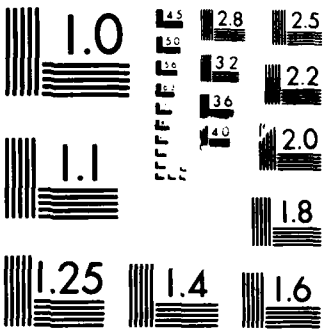
AD-A182 618 THE SOLUTION OF LARGE TIME-DEPENDENT PROBLEMS USING REDUCED COORDINATES(U) CALIFORNIA UNIV DAVIS DEPT OF CIVIL ENGINEERING K D MISH ET AL JUN 87 1/1

UNCLASSIFIED NCEL-CR-87 010 N68305-5345-3995 F/G 12/2 NL





(12)

CR 87.010
June 1987

NCEL
Contract Report

An Investigation Conducted By
University of California at Davis
Sponsored By Naval Facilities
Engineering Command

THE SOLUTION OF LARGE TIME-DEPENDENT PROBLEMS USING REDUCED COORDINATES

ABSTRACT This research is concerned with the idea of reducing a large time-dependent problem, such as one obtained from a Finite-Element discretization, down to a more manageable size while preserving the most important physical behavior of the solution. This reduction process is motivated by the concept of a projection operator on a Hilbert Space, and leads to the Lanczos Algorithm for generation of approximate eigenvectors of a large symmetric matrix. The proposed reduced coordinate algorithm is developed, compared to related methods, and applied to some representative problems in mechanics. Conclusions are then drawn, and suggestions made for related future research.

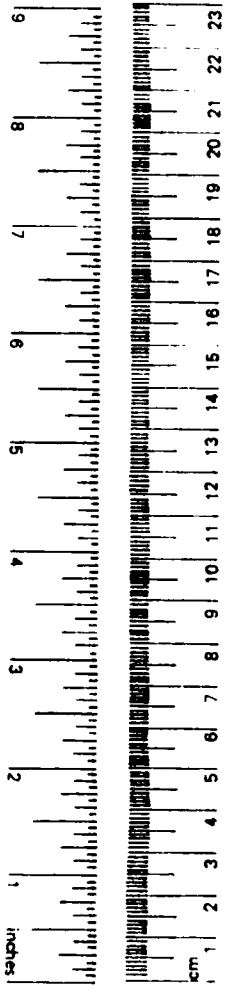
NAVAL CIVIL ENGINEERING LABORATORY PORT HUENEME CALIFORNIA 93043

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

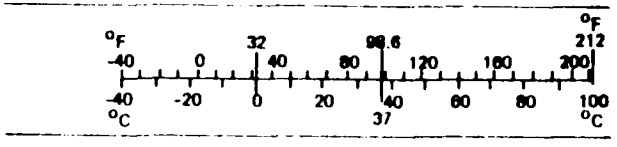
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2,000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1,000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CR 87.010	2. GOVT ACCESSION NO. AD-A182618	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Solution of Large Time-Dependent Problems Using Reduced Coordinates	5. TYPE OF REPORT & PERIOD COVERED Interim Mar 1986 - May 1987
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kyran D. Mish and Leonard R. Herrmann	8. CONTRACT OR GRANT NUMBER(s) N68305-5345-3995
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Civil Engineering University of California at Davis	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N YR023.03.01.008
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Civil Engineering Laboratory Port Hueneme, CA 93043-5003	12. REPORT DATE June 1987 13. NUMBER OF PAGES 104
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Facilities Engineering Command 200 Stovall Street Alexandria, VA 22332-2300	15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution is unlimited.	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) nonlinear finite element analysis, solution algorithm, reduced modal coordinates	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This research is concerned with the idea of reducing a large time-dependent problem, such as one obtained from a Finite-Element discretization, down to a more manageable size while preserving the most important physical behavior of the solution. This reduction process is motivated by the concept of a projection operator on a Hilbert Space, and leads to the Lanczos	

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 55 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Algorithm for generation of approximate eigenvectors of a large symmetric matrix. The proposed reduced coordinate algorithm is developed, compared to related methods, and applied to some representative problems in mechanics. Conclusions are then drawn, and suggestions made for related future research.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Contents

Chapter 1
Introduction

Chapter 1: Introduction

Overview of the Research

Many important problems in computational mechanics cannot be solved on modern sequential computers. Some of these unsolved cases include models with excessive memory requirements, ones that require inordinate amounts of computer time, and others that are simply numerically intractable. Some important problems suffer from more than one of these flaws. In many cases, technical improvements in computer architecture have reduced the size of the class of unsolved problems. A good example is the widespread use of virtual memory machines, which has eased physical memory requirements for many large problems. Similarly, parallel processing architectures will be used to solve other problems that require excessive computational cost on a sequential processor. But increasing the size of computer memory or increasing the number of processing units are not the only ways to achieve a satisfactory solution to a large, complex problem. Another useful method is to reduce the size of the problem so that the reduced model is small enough to solve on an appropriate computer, and yet the important engineering behavior of the model is preserved in the reduced problem.

Examples of this reduction process are abundant in engineering mechanics: any continuum problem that is solved via a discretization process involves the reduction from an infinite-dimensional problem to a finite-dimensional one. Examples of this simplification process include the Finite Element Method, Finite Difference Methods, and the use of truncated Fourier series. Many of these reduced problems are still too large to solve on modern computers, and so an attractive alternative is to find a way to reduce the size of the problem even further.

The research presented in this document represents an attempt to derive an algorithm for the solution of many large problems via a reduction in the number of independent solution coordinates. The mathematical principle underlying this reduction is that of a projection, and the algorithm is developed from this standpoint. Many of the basic principles of the proposed algorithm are widely used in engineering mechanics, and in that sense this research is not entirely new. However, many of the derivations and applications presented are quite different from those that have appeared in the literature, and represent an attractive alternative formulation for many otherwise computationally intractable problems.

The types of model reduction discussed in this research are very useful in the solution of many important engineering problems, but they are not intended to be a unified scheme to solve any large problem. There are undoubtedly many interesting models in mechanics that are not solvable using these coordinate reductions, and in these cases, the reduction should not be used. In many situations, the analyst will have sufficient physical intuition about the problem to be able to judge whether the reduction is warranted. In others, the proposed method can be used as a tool for preliminary analysis or design in conjunction with more expensive unreduced methods.

Finally, the use of reduced methods does not have to be confined to situations involving small computers or slow processors. The techniques developed in this document are appropriate for the solution of extremely large problems that are presently intractable on even the largest supercomputers. In addition, these coordinate reduction schemes often produce a mathematical problem that is more well-conditioned than a competitive unreduced formulation, and are thus more stable or accurate even in cases where the larger problem can be solved for a similar cost. In short, the research presented herein has the potential to become a useful and efficient part of the arsenal of numerical schemes that form the basis of modern computational methods in engineering.

Organization

This research is organized into a number of topics that comprise the development and application of a proposed reduced coordinate algorithm for the solution of large nonlinear problems in mechanics. It is assumed that the reader has some knowledge of the basic principles of mechanics, operator theory, and numerical analysis.

This first chapter casts the proposed research into an appropriate computational perspective, previews the contents of the other chapters, and establishes conventions used in this document.

The second chapter introduces the underlying mathematical theory for the proposed algorithm. The basic principles of projections onto a subspace are introduced, and set into the framework of the Projection Theorem, which is of fundamental importance in Applied Mathematics. Applications of the Projection Theorem are introduced, in both finite- and

infinite-dimensional settings. Next, the Fredholm Alternative Theorem is presented, along with applications and geometric interpretation. Finally, the development of Krylov methods for generation of the approximate spectrum of a linear operator are presented. In particular, the development and application of the Power Method and the Lanczos Algorithm are presented in this chapter.

The third chapter is devoted to the development of the proposed algorithm for the solution of large time-dependent problems using projection coordinates. The development is motivated by the cost of solving large approximate problems given by finite-element discretizations of time-dependent continuum boundary-value problems. In this setting, the finite-element approach leads to large systems of coupled ordinary initial-value problems, which are generally solved by a temporal integration scheme, such as Newmark's Method. This latter algorithm is developed in several different forms. Once the fully discretized initial-boundary-value problem is developed, the proposed algorithm is shown to be a natural way to solve these problems using projection methods. In this section, this reduced coordinate algorithm is examined in light of computational considerations, and is compared to other established methods in the setting of the theory developed to this point.

The fourth chapter consists of the application of the reduced coordinate algorithm to some representative problems in solid mechanics. In each case, the proposed algorithm is compared to direct (unreduced) solution schemes on the basis of complexity and computational effort.

Finally, the fifth chapter is devoted to general conclusions and proposed avenues for further study in the field of reduced coordinate models. The list of references for the document follows this chapter.

Conventions

There are a number of important symbolic conventions that will be used in this document in order to simplify the requisite notation. Any exceptions will be noted whenever an inconsistent nomenclature is introduced. For the most part, the following conventions will be used:

(1) scalars will be denoted by lower-case Greek letters, such as α, β_{j+1}. The scalar components of a vector or matrix will typically be written as subscripted Greek letters.

(2) vectors will be denoted by lower-case Roman letters, such as q_j, v. Columns of matrices are vectors, and will usually be the subscripted lower-case version of the same letter used for the matrix.

(3) matrices will be denoted by upper-case Roman letters, such as A or Q_j. Symmetric matrices will generally be represented by symmetric letters like A or T, but established conventions may violate this rule. A subscripted matrix like Q_j may be used to emphasize unusual matrix dimensions.

(4) operators will also be represented by upper-case Roman letters, just like matrices. In the case of finite-dimensional operators, this is a natural convention, since these operators can easily be identified with their corresponding matrix. For infinite-dimensional operators, the context will be adequate to avoid confusion.

(5) vector spaces will be denoted by **bold** upper-case Roman Letters, such as **H** or $\mathbf{M}^{\perp}$. Another use for these characters will be to denote domains of functions, which are often vector spaces.

Figures, equations, tables, and definitions will all be listed in terms of the chapter number, and then by order within the chapter. All chapters will have their own page numbering scheme, with the number to the left of the decimal point indicating the chapter, and the page number occurring after the decimal point.

Finally, references will be presented, in parentheses, by the last name of the primary author and the reference date, e.g., (Lanczos, 1950). References will be collected at the end of the work, listed in alphabetical order.

Chapter 2

Mathematical Preliminaries

Chapter 2: Mathematical Preliminaries

Introduction

The basic principle underlying this research is that of a projection of some complicated problem onto a simpler setting. The precise mathematical formulation of this idea involves the concept of a projection operator on a Hilbert space. This topic may seem recondite to most engineers, but can in fact be understood in terms of the generalization of the topological and algebraic properties of ordinary three-dimensional space. The first part of this chapter develops the required theory from exactly this standpoint, and culminates in the Projection Theorem, which is one of the most important results of modern applied mathematics, and which will be used often in subsequent sections of this research.

Once the concept of a projection operator on an abstract space is defined, the mathematics of spectral theory can be developed to explore the idea of which projections are appropriate for a given problem. The second part of this chapter is devoted to this line of reasoning, and results in the Fredholm Alternative Theorem. This theorem classifies which problems are solvable and also demonstrates the importance of different parts of the spectrum in the solution of the reduced problem.

The spectrum of an operator can be used to determine appropriate subspaces for projection solutions of the large problems that arise in Finite-Element modelling, but the actual construction of these subspaces is an iterative process that must be implemented on the computer. One of the most useful classes of numerical algorithms for generating these approximating subspaces can be developed by considering a special subspace called the Krylov Subspace. Two particular schemes for constructing approximations based on this Krylov Subspace, namely the Power Method and the Lanczos Algorithm, are derived at the end of this chapter.

The Definition of a Hilbert Space

There are several concepts that we associate with vectors in ordinary three-dimensional Euclidean space, $\mathbf{R}^3$. The first of these is the concept of length or norm, which can be expressed in terms of the distance between a given vector v and the zero vector o. In $\mathbf{R}^3$, the length of a vector v with components v_1, v_2, and v_3 is given by:

$$\| v \| = (v_1^2 + v_2^2 + v_3^2)^{1/2} = \langle v, v \rangle^{1/2} \qquad \text{Eqn 2.1}$$

where $\langle v, v \rangle$ is the usual inner product for $\mathbf{R}^3$, namely $\langle v, v \rangle = v^T v$. This length in $\mathbf{R}^3$ is thus the square root of the inner product of a vector with itself, and this concept of inner product is another fundamental topological property that we associate with three-dimensional space $\mathbf{R}^3$. In addition to providing the notion of length, the inner product also defines the angle between two vectors, via the relation:

$$\langle u, v \rangle = \| u \| \| v \| \cos(\theta_{uv}) \qquad \text{Eqn 2.2}$$

Finally, our physical intuition of space tells us that it has no "holes", in that any convergent sequence of vectors in $\mathbf{R}^3$ tends to a limit vector that is also contained in $\mathbf{R}^3$. In an abstract setting, this property is termed completeness, and this term agrees with the intuitive connotation of "complete" as meaning "whole" or "entire".

In summary, we note that $\mathbf{R}^3$ is a complete normed inner-product space, and the norm (length) function is defined in terms of the inner product. These concepts characterize our physical intuition about $\mathbf{R}^3$, and they are easily generalized to n-dimensional Euclidean space $\mathbf{R}^n$, or to abstract linear vector spaces S of arbitrary dimension. It is exactly these concepts which underlie the definition of a Hilbert Space:

> Definition 2.1: A Hilbert Space is a complete normed linear vector space with the norm induced by an inner product.

The main concept to remember about a Hilbert space is that it is an attempt to generalize the topological behavior of familiar three-dimensional space $\mathbf{R}^3$ to vector spaces of higher

dimension. In particular, because a Hilbert space is endowed with an inner product, the concept of the angle between two vectors (recall Eqn 2.2) is preserved. In particular, the concept of "perpendicularity" is generalized:

Definition 2.2: Two vectors u and v in a Hilbert space **H** are termed orthogonal if their inner product vanishes (i.e., $\langle u, v \rangle = 0$)

A subset **M** of a linear vector space **S** may be a vector space in its own right. In this case, the subset is referred to as a subspace, or a linear manifold (often abbreviated to simply "manifold" when the context of linearity is clear):

Definition 2.3: A set **M** in a linear vector space **S** is termed a subspace (or a linear manifold) if, for all vectors u and v in **M**, and for all scalars α and β, the vector $(\alpha u + \beta v)$ is also in **M**. (Some concrete examples of finite-dimensional manifolds are shown in Figure 2.1)

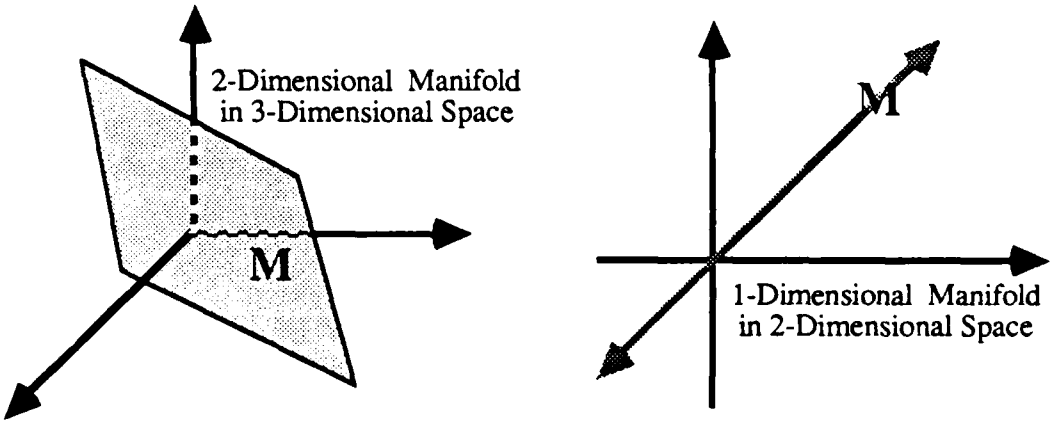


Figure 2.1: Examples of Linear Manifolds

Note that a linear manifold necessarily contains the zero vector (choose $\alpha = \beta = 0$), and thus "inherits" an identity as a vector space from the parent space **S**. If the parent space is complete, and the manifold (considered as a vector space) is also complete, then the manifold is referred to as closed. Two important classes of manifolds are always closed: finite-dimensional manifolds, and manifolds that are "perpendicular" to another manifold. This latter case is important enough to warrant a definition:

Definition 2.4: If **M** is a linear manifold in a Hilbert space **H**, then the set $\mathbf{M}^{\perp}$ (the linear manifold orthogonal to **M**) is defined as consisting of all vectors in **H** that are orthogonal to every vector in **M**. This set $\mathbf{M}^{\perp}$ is termed "M-perp", to indicate that the whole manifold is perpendicular to **M**.

Another term for $\mathbf{M}^{\perp}$ is the orthogonal complement of **M**. Figure 2.2 illustrates the orthogonal complements $\mathbf{M}^{\perp}$ for the types of manifolds shown in Figure 2.1.

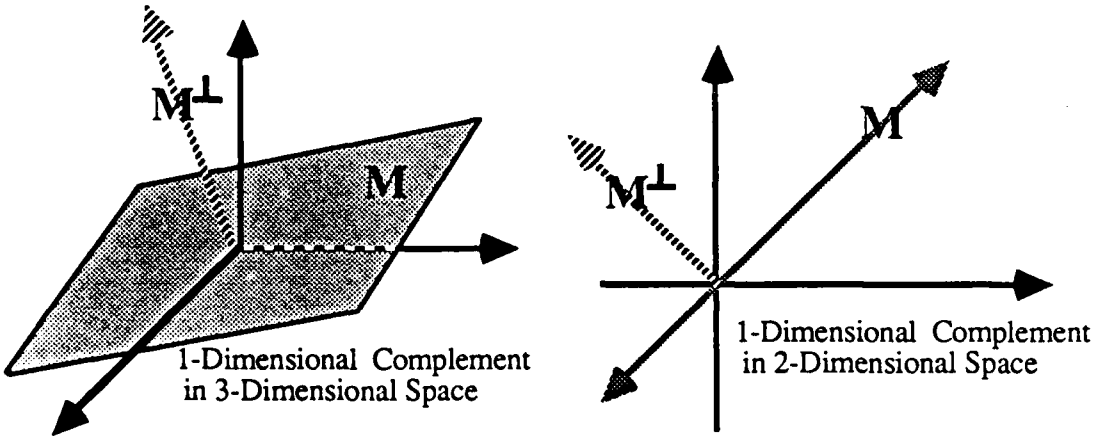


Figure 2.2: Examples of Orthogonal Complements

The Projection Theorem

Closed linear manifolds in a Hilbert space are the setting of the following theorem, which is among the most important in Applied Mathematics (Stakgold, 1979):

Theorem 2.1 - The Projection Theorem. Let **M** be a closed linear manifold in a Hilbert space **H**. Every vector u in **H** can be expressed uniquely as the sum

$$u = v + w$$

where v is a vector in **M** and w is a vector in $\mathbf{M}^{\perp}$. The vector v is termed the orthogonal projection (or simply the projection) of u on **M**, and the vector v can be characterized as the unique vector in **M** that is closest to u.

It is important to note the terms "orthogonal" and "closest" in the statement of this theorem. The setting of the Projection Theorem is a Hilbert Space, whose distinguishing feature is its inner product function. This inner product induces the concept of angle between vectors in the space, including the important special case of perpendicularity between vectors. The concept of "closeness" of the vector v to the vector u is defined in terms of the norm of the error u - v, and this norm is given by the inner product as well. The "closest" vector v to u is the one that minimizes the norm of the error u - v. Thus the terms "orthogonal" and "closest" have precise meanings in terms of the underlying inner product on the Hilbert Space **H**, and an application of the Projection Theorem in practice means that, on some level, a minimum-norm problem is being solved. This concept will appear again and again throughout this document. The geometric interpretation of these ideas for three-dimensional space is shown in Figure 2.3.

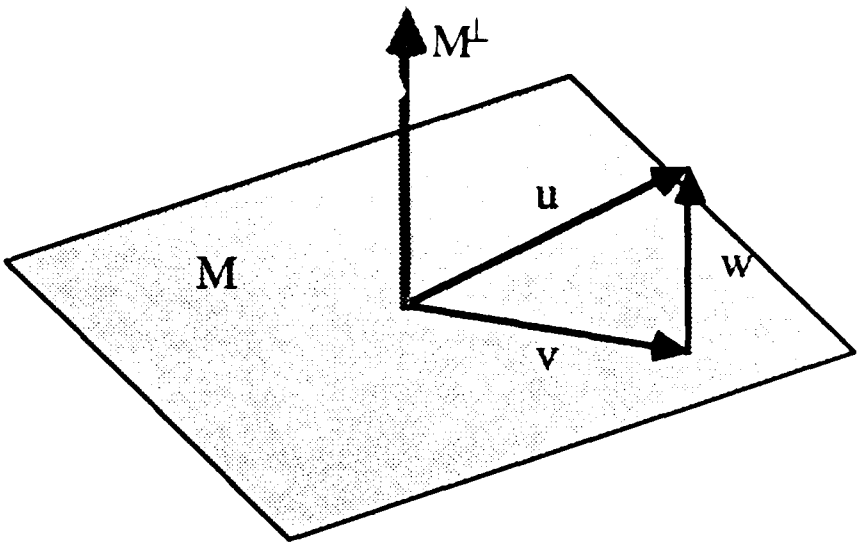


Figure 2.3: Interpretation of the Projection Theorem

Given a closed linear manifold **M**, a Projection Operator P can be defined by the action of P on any arbitrary vector in **H**. If u = v + w is a vector in **H**, and the vector v is the orthogonal projection of u onto **M** (whose existence and uniqueness is guaranteed by the Projection Theorem), then the projection operator P can be implicitly defined by its effect on u, namely Pu = v. This operator merely projects any vector in the space onto the manifold **M**, which results in an error w = (I - P)u, where I is the identity operator. This

last relation also defines the projection operator for $\mathbf{M}^{\perp}$, namely (I - P). Thus the decomposition of u into orthogonal components v and w can be written in terms of projection operators:

$$u = v + w = Pu + (I - P)u \qquad \text{Eqn 2.3}$$

Note that any projection operator satisfies $P^k = P$ for $k > 0$, since the projection onto the subspace need only be done once. (Such operators are termed idempotent).

The Projection Theorem has a geometrical interpretation that is important in the application of approximation theory. The vector u is to be approximated by some vector v in **M**, and this introduces an error vector e = u - v. This approximation v and error e implicitly define two manifolds, as shown in Figure 2.4:

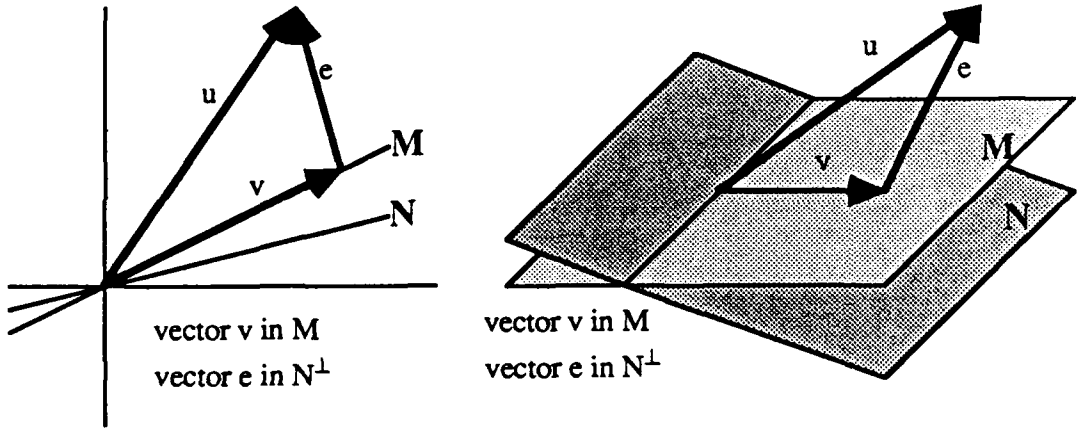


Figure 2.4: Approximation Manifolds

(1) The manifold **M** containing the approximation v. In the problem of the approximation of functions, this manifold is often spanned by a basis of interpolating functions, so **M** is sometimes called the space (subspace) of "basis" functions.

(2) The manifold that is perpendicular to the error vector e. This manifold will be termed **N**, and will be associated with the projection operator Q. Every choice of an independent vector w from **N** gives an equation $\langle e, w \rangle = 0$ that can be interpreted as a "test" for a vanishing error component. From this interpretation, the manifold **N** is often termed a space of "test" functions.

The general problem involves the approximation of u by v, and the criterion for the selection of this approximation v is the projection relation $Q(e) = Q(u - v) = 0$. This type of formulation is often encountered in applications (the weak formulations of Lagrangian Mechanics that lead to the Finite Element Models in Chapter 3 and 4, for example), and there are many schemes for choosing the manifolds **M** and **N**. What is important here is the realization that the Projection Theorem tells how to choose **N** in order to minimize the norm of the error u - v. This minimum-error solution is obtained by taking e = u - v perpendicular to the manifold **M**, which is equivalent to choosing **N** = **M**. In this case, the space of basis functions and the space of test functions coincide, and the error is minimized in the natural norm for the problem. This type of approximation is termed a Galerkin Approximation, and such approximations are obviously Projection Solutions, in that they involve the projection of the problem onto an approximating subspace.

Projections Onto a Subspace (Matrix Case)

Let $x \in \mathbf{R}^n$, $y \in \mathbf{R}^m$, and $Q \in \mathbf{R}^{n \times m}$, with $m \leq n$. When we write x = Qy, we are saying that x is a linear combination of the columns of Q, or that x lies in the column space of Q, which is denoted CS(Q). To see this fact, simply partition the matrix Q into its columns

$[q_1, q_2, ..., q_m]$, the vector y into rows $(\psi_1, \psi_2, , \psi_m)^T$, and form the product:

$$Qy = \psi_1 q_1 + \psi_2 q_2 + ... + \psi_m q_m$$

Thus x = Qy is just a linear combination of Q's columns. This provides a convenient shorthand for expressing any vector in CS(Q): as the elements of y vary over all real numbers, the vector x = Qy ranges over all the vectors in CS(Q). We will assume that the dimension of CS(Q) is m, so that Q's columns are linearly independent, and furthermore, that we have orthogonalized and normalized Q's columns so that $Q^TQ = I_m$, the identity matrix of order m. Note that $QQ^T \neq I_n$, unless m = n, since the rank of the product of two matrices cannot be larger than the rank of either one, and rank(Q) = dim(CS(Q)) = m.

With this convention, we see that CS(Q) forms an m-dimensional linear manifold of $\mathbf{R}^n$, and that Q's columns form an orthonormal basis for this manifold. In this case, the Projection Theorem implies that any vector w can be decomposed into a component u that

lies in CS(Q), and a component v that is in $CS(Q)^{\perp}$, the orthogonal complement of the column space of Q, as in Eqn. 2.3:

$$w = u + v \qquad u \in CS(Q),\ v \in CS(Q)^{\perp} \qquad \text{Eqn 2.4}$$

The vector u is given by QQ^Tw, and the vector v by $(I - QQ^T)w$. By comparing Equations 2.3 and 2.4, we can see that the nxn matrix QQ^T is a projector onto CS(Q), and $(I - QQ^T)$ is a projector onto $CS(Q)^{\perp}$. A geometric picture for the cases n = 2 and n = 3 is shown in Figure 2.5.

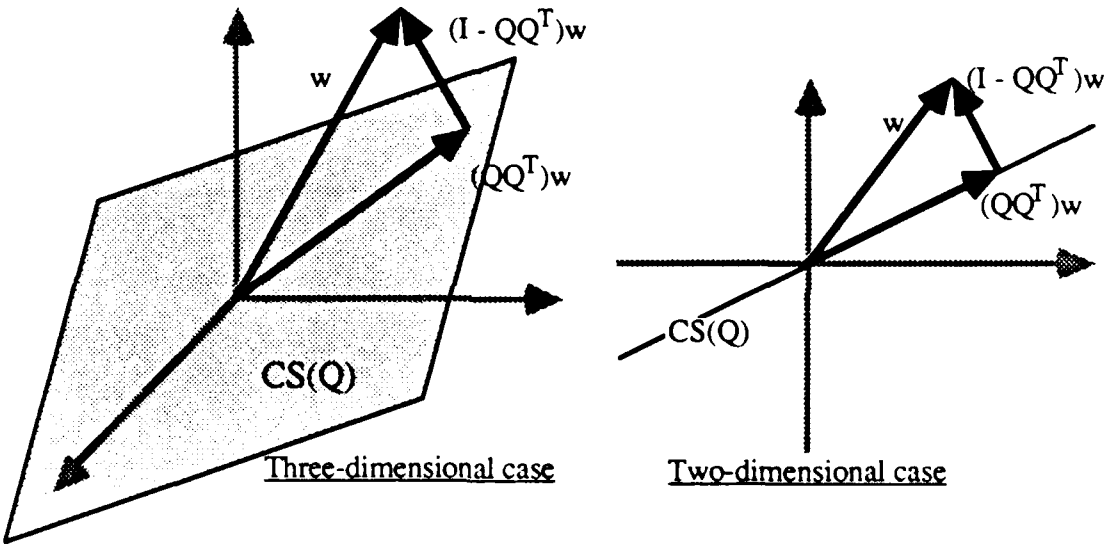


Figure 2.5: Examples of Vector Projections

If A is an nxn symmetric matrix and Q a matrix with m orthonormal columns, then an approximate solution x of the system $Ax = f$ can be sought in the column space of Q by solving the problem $AQy = f$. Unfortunately, this problem has n equations and m unknowns, and thus is inconsistent, in general. The projection of this problem onto CS(Q) could be obtained by multiplying by the projector QQ^T to give the matrix equation $QQ^TAQy = QQ^Tf$, but this is an mxm problem embedded in n-dimensional space. A better approach would be to simply multiply the equation $AQy = f$ by Q^T to obtain the desired

relation $Q^TAQx = Q^Tb$, which is an mxm system in m-dimensional space. This is a Galerkin approximation, since the approximate solution space CS(Q) is the same as the projection space.

It is appropriate at this point to conjecture on the criteria that would make the CS(Q) a "good" approximate solution space. At the very least, we would like to satisfy:

(1) Q^TAQ has to somehow be a "good" approximation to A

(2) Q^TAQ should have some simple structure (i.e. banded, triangular, etc.)

(3) It should be economical to form Q, or to add vectors to Q to increase the rank (and hence the accuracy) of the projected matrix Q^TAQ.

Solvability Conditions

If we are going to solve operator equations like $Ax = f$ via projection methods, then some consideration must be given to the problem of determining the conditions under which the operator equation is solvable. These solvability conditions can be phrased in terms of inner products, and the concept of projection operators can be used to generate approximate solutions of $Ax = f$ even when the solvability conditions are not satisfied. In the following development, the operator A will be taken to be self-adjoint, in that $\langle Au, v\rangle = \langle u, Av\rangle$. Extensions of this theory to the non-self-adjoint problem can be made, but they are not needed at the present stage of this research, and the nomenclature gets more complicated.

When the operator A is defined over a domain in a Hilbert Space, the set of all vectors in this domain that satisfy $Av = 0$ is called the Null Space of the operator A, and is denoted by NS(A). The set of all vectors $y = Ax$ is called the Range of A. (We will be concerned with operators whose domain and range are both subsets of the same Hilbert Space **H**.) If the operator equation $Ax = f$ is to be solvable, then the vector f must be in the range of A, so that, at the simplest level, a solvability condition is merely any condition that characterizes the range of A. If the equation $Ax = f$ is solvable, then the inner product equation

$$\langle Ax, v\rangle = \langle f, v\rangle$$

is satisfied for any vector v, as long as x is the desired solution. In particular, if v is in NS(A), then:

$$\langle f, v\rangle = \langle Ax, v\rangle = \langle x, Av\rangle = \langle x, 0\rangle = 0$$

since A is self-adjoint, and Av = 0 for any v in NS(A). This relation places an important condition on the vector f: if the self-adjoint operator equation Ax = f is to have a solution, then a necessary condition for this solution to exist is that the inner product <f, v> vanish for any vector v in the null space of A. In the terminology of the last section, if P is the projection operator onto the null space of A, then Pf = 0 is a necessary condition for a solution of Ax = f to exist. If this condition is satisfied and NS(A) is not empty, then the solution of Ax = f will be non-unique. In that case, if x is a solution, then

$$A(x + v) = Ax + Av = f + 0 = f$$

so that x + v is also a solution for any v in NS(A).

An obvious question arises: what if the equation Ax = f has to be solved, and yet f is not orthogonal to the null space of A? Can we "set our sights lower" and find some approximate solution to a related problem? This question can be answered simply here. Consider the new problem obtained by projecting Ax = f onto the orthogonal complement of NS(A):

$$(I - P)Ax = (I - P)f \quad \text{or} \quad Bx = g$$

Here, the operator B = (I - P)A is the projection of A onto this complement space, and the vector g = (I - P)f is the projection of f onto the same space. It is easy to see that B and A have the same null space, so the necessary condition for Bx = g to be solvable is that <g, v> = 0 for any vector in NS(A). Since g is orthogonal to NS(A) by construction, this condition is trivially satisfied. Thus it appears that the approximate problem Bx = g can be solved, though in order for the approximation to be accurate, the error Pf = f - g in the right-hand side must somehow be unimportant.

The Spectrum of an Operator

There is an extremely important type of null space that arises in applications. The operator A is taken (as usual) to be self-adjoint, and the null space of the operator $(A - \lambda I)$ is sought, where λ is some real number. The set of all numbers λ such that the operator equation

$$(A - \lambda I)u = 0$$

has non-trivial solutions is called the Point Spectrum of A. (If A is a matrix operator, the

point spectrum is often simply called the spectrum of A). Any such number λ in the point spectrum is termed an eigenvalue, and the corresponding nontrivial solution u is called an eigenvector; the pair (λ, u) is termed an eigenpair. For a finite-dimensional operator A, the set of eigenvectors of A forms a basis for the underlying (finite-dimensional) Hilbert Space, and can be chosen to form an orthogonal set. Henceforth, when we speak of the eigenvectors of a self-adjoint operator A we will assume that this orthogonalization has already been performed, and in fact that all the eigenvectors have been normalized to unit length. Then the set of eigenvectors of A can be taken as an orthonormal basis for the underlying Hilbert Space **H**.

It is worthwhile to examine the solution x of the n-dimensional operator equation $Ax = f$ in the light of the existence of an orthonormal set of eigenvectors of A. If we denote these eigenvectors by u_i, and the corresponding eigenvalues by λ_i, then both the solution x and the data f can be expanded in terms of the u_i:

$$x = \sum_{i=1}^{n} \xi_i u_i \qquad f = \sum_{i=1}^{n} \eta_i u_i$$

With these expansions, the equation $Ax = f$ can be written in terms of A's eigenvectors:

$$Ax = A \sum_{i=1}^{n} \xi_i u_i = \sum_{i=1}^{n} \lambda_i \xi_i u_i = \sum_{i=1}^{n} \eta_i u_i$$

The identification of each component of the sum gives the coefficients of the solution x:

$$\xi_i = \eta_i / \lambda_i \qquad \text{Eqn 2.5}$$

This is an extremely important result: the coefficients of the solution of $Ax = f$ occur in inverse ratio to the eigenvalues of A. This means that, for a general distribution of the coefficients of f, the most important eigenvalues in terms of approximating the solution x are the ones with the least magnitudes, since they contribute the largest effect on the components of x. In other words, given a random distribution of vectors f, an approximation to the solution of $Ax = f$ based on the eigenvectors associated with the minimal eigenvalues (in modulus) would be expected to give more accurate results than one based on the eigenvectors associated with the maximal eigenvalues. This bias towards the minimal eigenvalues would become more pronounced as the ratio of the moduli of the extreme eigenvalues $|\lambda_n/\lambda_1|$ becomes large, where λ_1 and λ_n are the minimum and

maximum eigenvalues, respectively (in terms of their magnitudes). This ratio $|\lambda_n/\lambda_1|$ represents the "spread" of the eigenvalues of A, as shown in Figure 2.6:

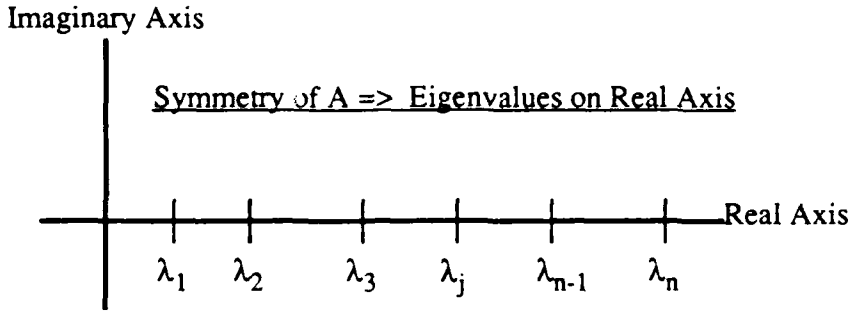


Figure 2.6: Eigenvalues for Positive-Definite Matrix A

If the ratio $|\lambda_n/\lambda_1|$ is large, the transformation A induces a wide range of characteristic "scales" on the data f and the solution x, and this existence of "multiple scales" may cause numerical problems for finite-precision calculations involving the matrix A. For this reason, this ratio is also known as the (2-norm) Condition Number of the matrix A, since whenever this number is large, the matrix is potentially ill-conditioned.

Finally, the maximal eigenvalue λ_n defines a norm on the set of nxn symmetric matrices A that is termed the spectral radius $\rho(A)$. This norm is the natural extension of the norm defined on $\mathbf{R}^3$ by Equation 2.1, in that it measures the stretch induced by the transformation A on vectors in $\mathbf{R}^n$, because it is equivalently defined by:

$$\rho(A) = \lambda_n = \max (\| Ax \| / \| x \|)$$

where the maximum is taken over all x in $\mathbf{R}^n$, and $\| x \|^2 = x^T x = \langle x, x \rangle$.

The Fredholm Alternative Theorem

If (λ, u) is an eigenpair of A, then it is easy to see that the null space of $(A - \lambda I)$ is not empty, since at least it contains u. In this case, the operator equation $(A - \lambda I)x = f$ must satisfy the solvability condition $\langle f, u \rangle = 0$. If λ is not an eigenvalue of A, then the null

space of $(A - \lambda I)$ is empty by definition, and so there are no solvability conditions that need to be placed on f. Thus the behavior of the operator equation $(A - \lambda I)x = f$ depends very strongly on whether the parameter λ is an eigenvalue or not. These two alternatives are contrasted in the following result, which is known as the Fredholm Alternative Theorem. The statement of the theorem below is similar to that found in (Noble, 1977), and it holds only for finite-dimensional Hilbert Spaces. A similar result can be stated for so-called compact operators on an infinite-dimensional Hilbert Space, but the definition of compactness is beyond the scope of this section. The important interpretations of this theorem for the purposes of this research hold for all the Hilbert Spaces that underlie this research (namely the infinite-dimensional solution space for the continuous physical problem, as well as the finite-dimensional vectors spaces populated by the Finite Element matrices that will be discussed in Chapters 3 and 4).

Theorem 2.2 - The Fredholm Alternative Theorem. Let $(A - \lambda I)x = f$ be a set of n linear equations in n unknowns, where A is a symmetric matrix and λ a given real number. Then exactly one of the following alternatives is true:

(1) $\lambda = \lambda_i$, where λ_i is an eigenvalue of A, in which case a solution x exists if and only if the condition $\langle f, u_i \rangle = 0$ holds for every eigenvector u_i associated with the eigenvalue λ_i. In this case, infinitely many solutions exist, each of the form $x + \alpha u_i$, where α is an arbitrary real number and u_i is any eigenvector associated with the eigenvalue λ_i.

(2) λ is not one of the eigenvalues of A, in which case the unique solution x can be written as :

$$x = \sum_{i=1}^{n} \{\langle u_i, f \rangle / (\lambda - \lambda_i)\} u_i$$

where (λ_i, u_i) are the eigenpairs of A and the u_i are taken to be an orthonormal set.

This relation is important because it sums up the conditions under which the equation Ax=f has a solution, and also because it shows how the solution depends on the eigensystem of

A, as well as the data f and the parameter λ. It is easy to see that the expansion of x is most dependent upon the eigenvalues that are closest to the parameter λ. In mechanical applications, this scalar λ has the interpretation of a frequency, and the dependence of the solution on the nearest eigenvalues leads to the familiar phenomenon of resonance.

The Power Method

Probably the most commonly known method for finding a part of the spectrum of a matrix is the Power Method. Under the right conditions it can be used to find exactly one eigenpair of a matrix A, namely the eigenvalue of largest modulus and the associated eigenvector (often termed the dominant eigenpair). A first draft of an algorithm for the implementation of the Power Method looks like:

Algorithm 2.1: Power Method

Finds: the dominant eigenpair (λ_n, u_n) of a matrix A

Given: initial vector x_0

Repeat for i = 1, 2, 3....

$x_i = Ax_{i-1}$

$\theta_i = \| x_i \| / \| x_{i-1} \|$

until converged

The pair (θ_i, x_i) is an approximation to (λ_n, u_n)

In practice, the algorithm would contain a normalization step, to keep the length of the vectors x_i from becoming too large or too small. This step is omitted from the algorithm to call attention to an important fact: the sequence of vectors $\{x_1, x_2, ..., x_i\}$ is identical to the sequence $\{Ax_0, A^2x_0, ..., A^ix_0\}$. This latter sequence of vectors is called the Krylov Sequence associated with the matrix A and the initial vector x_0. The subspace spanned by this sequence of vectors is called the Krylov Subspace, which will be shown to play an

important role in the Lanczos Algorithm. The Krylov subspace will be denoted by:

$$\mathbf{K}(A, u, k) = \text{span}\{Au, A^2u, A^3u, ..., A^ku\} \qquad \text{Eqn 2.6}$$

With this convention, it is clear that the Power Method is just a way to generate a Krylov sequence, and the convergence of the Power Method is equivalent to the convergence of the Krylov sequence to the dominant eigenvector of the matrix A. What should be noted is that only the last member of the sequence is used in the Power Method, since all the earlier terms are merely a means to the end of finding one eigenpair. A logical question to ask would be whether several (or all) of the terms of the sequence could be used to find estimates for more than one of A's eigenpairs. In fact, the entire subspace $\mathbf{K}(A, u, k)$ can be used to calculate estimates of exactly k eigenpairs of the matrix A. This fact is the basis for the Lanczos Algorithm.

The convergence of the Power Method is easy to establish, as long as A has a single dominant eigenpair (i.e., λ_n is a simple eigenvalue), A is non-defective, and x_0 is not orthogonal to the associated dominant eigenvector u_n. In this case, the vector x_0 can be expressed in terms of the eigenvectors of A as:

$$x_0 = \sum_{i=1}^{n} \xi_i u_i \qquad Ax_0 = \sum_{i=1}^{n} \lambda_i \xi_i u_i \qquad A^p x_0 = \sum_{i=1}^{n} \lambda_i^p \xi_i u_i = \lambda_n^p \sum_{i=1}^{n} \frac{\lambda_i^p}{\lambda_n^p} \xi_i u_i$$

As p goes to infinity, the ratio $(\lambda_i/\lambda_n)^p$ goes to zero in every case except when i = n, so that the Krylov Sequence converges to the dominant eigenvector u_n.

The Power Method can be applied with a shift σ by replacing the matrix A with the shifted matrix $(A - \sigma I)$. In this case, the algorithm converges to the eigenpair that maximizes the quantity $(\lambda_i - \sigma)$, as long as this eigenvalue is simple and the starting vector is not orthogonal to the associated eigenvector. In order to find the minimal eigenpair (the one associated with the eigenvalue of least modulus), the Power Method can be applied in an inverse setting by replacing A with its inverse A^{-1}. This is accomplished in practice by replacing the step "$x_i = Ax_{i-1}$" in Algorithm 2.1 by the inverse step "solve $Ax_i = x_{i-1}$". Finally, the algorithm can be applied in an inverse setting with a shift σ by solving the system $(A - \sigma I)x_i = x_{i-1}$ for the iterate x_i. In this case, the convergence of the iteration can

be established (under the usual conditions given above) to the eigenpair that minimizes the quantity $(\lambda_i - \sigma)$. This eigenpair is the one whose eigenvalue is closest to the shift point σ. This last case is of great utility in applications involving resonance, since the shift point can then be interpreted as a frequency component of a time-dependent forcing term. In this case, the eigenpair found by the Power Method is one likely to be excited by resonance effects (at least in the linear case, where sub- and super-harmonic resonances are not an issue).

The Lanczos Algorithm

The Lanczos Algorithm is an iterative method that can be used to find relatively accurate estimates of some of the extremal eigenvalues of a symmetric matrix A, along with good estimates of the associated eigenvectors. It is used in this research as a way to find suitable approximating subspaces (such as the set of converged eigenvectors from the Lanczos iteration) for the projection solution of problems involving a large symmetric matrix A.

There are many starting points for a derivation of the Lanczos Algorithm. The discussion of the last section arrived at the Lanczos Algorithm as a generalization of the Power Method, where the entire Krylov Subspace is used for the approximation manifold. Alternatively (Golub, 1985), the Lanczos Algorithm can be derived from considerations of the optimization of the Rayleigh Quotient , which is defined by:

$$R(A, x) = (x^T A x)/(x^T x) \qquad \text{Eqn 2.7}$$

The Rayleigh Quotient is a real-valued function of a vector (a functional), which satisfies:

$$\min R(A, x) = \lambda_1 \quad \text{and} \quad \max R(A, x) = \lambda_n \qquad \text{Eqn 2.8}$$

where λ_1 and λ_n are the minimal and maximal eigenvalues of A , respectively. (As usual, it is assumed that A is symmetric.) The min and max are taken over all nx1 vectors x. This functional can easily be extended to infinite-dimensional linear self-adjoint operators by replacing the minimum and maximum with the infimum and supremum, respectively.

If x is taken to lie in CS(Q) (i.e., $x = Qy$) instead of $\mathbf{R}^n$, then the equalities will generally not be obtained:

$$\min R(A, Qy) \geq \lambda_1 \quad \text{and} \quad \max R(A, Qy) \leq \lambda_n$$

An alternative way to express these terms is to note that:

$$R(A, Qy) = (Qy)^T AQy/(Qy)^T Qy = y^T Q^T AQy/y^T y = R(Q^T AQ, y)$$

where y is an mx1 vector. The Rayleigh Quotient for the vector $x = Qy$ is merely the Rayleigh Quotient for the reduced matix Q^TAQ. Henceforth, when discussing the minimum or maximum of R(A, x), the dependence on the vector x will be suppressed, so that R(A, x) will be written as R(A).

A natural way to measure the accuracy of Q^TAQ as an approximation to A is to try to optimize the Rayleigh Quotient, in the sense that Q is to be chosen to give the least minimum and the largest maximum. Alternatively, we could consider the situation when the nxm matrix Q_m is known, and it is desired to append a vector q_{m+1} to it to obtain a new matrix Q_{m+1} that gives a better projection approximation to A, in the sense that:

$$\min R(Q_{m+1}^T AQ_{m+1}) < \min R(Q_m^T AQ_m) \qquad \text{Eqn 2.9.a}$$

$$\text{and } \max R(Q_{m+1}^T AQ_{m+1}) > \max R(Q_m^T AQ_m) \qquad \text{Eqn 2.9.b}$$

Since R(A, x) is a scalar-valued function of a vector x, a logical way to achieve this type of progress is to take a "steepest-descent" approach, by choosing the new vector q_{m+1} to have a component in the direction of the gradient of R(A, x):

$$\text{Grad}(R(A,x)) = d((x^TAx)/(x^Tx))/dx = 2[Ax - R(A, x)x]/(x^Tx)$$

Since $2[Ax - R(A, x)x]/(x^Tx)$ is in the span of {x, Ax}, we can satisfy Equations 4.3 by requiring that q_{m+1} contain components in the directions of $Aq_1, Aq_2, \ldots, Aq_m$. This observation leads to the following iteration scheme for the generation of an approximation $Q_m^TAQ_m$ to the matrix A:

Given an initial vector q_o and a symmetric matrix A:

Let $q_1 = Aq_o$

Find q_2 in span $\{q_1, Aq_1\}$

Find q_3 in span $\{q_1, Aq_1, q_2, Aq_2\}$ = span $\{q_1, Aq_1, Aq_2\}$ = span $\{q_1, Aq_1, A^2q_1\}$

.....

.....

.....

Find q_m in span $\{q_1, Aq_1, A^2q_1, ..., A^{m-1}q_1\} = \mathbf{K}(A, q_o, m)$

The optimization of the Rayleigh Quotient leads to the choice of the Krylov Subspace as the approximating subspace for a projection solution involving the matrix A. If the columns of Q form an orthonormal basis for the Krylov Subspace, then the matrix Q^TAQ is an optimal (in the sense of the Rayleigh Quotient) approximation to A. Thus, we are led to the task of finding an orthonormal basis for the Krylov Subspace.

In theory, the standard way to turn a sequence of linearly independent vectors such as $q_1, Aq_1, A^2q_1, ... A^{m-1}q_1$ into an orthonormal set is to apply the Gram-Schmidt Orthogonalization procedure. In practice, this procedure is numerically unstable, and the variations on this scheme (such as "Modified Gram-Schmidt", which reorganizes the computations somewhat to diminsh the effect of cancellation of significant digits) that will produce a set of orthogonal vectors are too expensive to implement for large m. (Recall that we expect $m \ll n$, but since n can easily be on the order of hundreds of thousands, m can still become "very large".) Some other way of producing orthonormal bases for the Krylov Subspace must be found. Luckily, for a symmetric matrix A, the vectors in the Krylov sequence $Au, A^2u, ..., A^nu$ satisfy a three-term recurrence (see Mish, 1987, Chapter 4, for details). This means that Q^TAQ, the projection of A onto the Krylov Subspace, is a tridiagonal matrix T, so that $Q^TAQ = T$ or, equivalently, $AQ = QT$.

To make this last relation more concrete, define the matrix $T = \text{tridiag}(\beta_i, \alpha_i, \beta_{i+1})$, and write the three term recurrence by columns, so that:

$$AQ = QT \quad \text{or} \quad A\,[q_1, q_2, \ldots, q_n] = [q_1, q_2, \ldots, q_n] \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \beta_2 & \alpha_2 & \beta_3 & 0 & \\ 0 & \beta_3 & \alpha_3 & \beta_4 & \cdots \\ \vdots & 0 & \cdots & \cdots & \beta_n \\ 0 & \cdots & 0 & \beta_n & \alpha_n \end{bmatrix}$$

This recurrence can be used as an iteration scheme for constructing the desired orthonormal basis for the Krylov Subspace. If (for purposes of simplifying the three-term recurrence) q_0 is defined as the zero vector, and $\beta_1 = 1$, then this matrix equality can be written columnwise as:

$$Aq_j = \beta_j q_{j-1} + \alpha_j q_j + \beta_{j+1} q_{j+1} \qquad (j = 1, 2, \ldots, n-1) \qquad \text{Eqn 2.10}$$

This relation can be solved for $\beta_{j+1}q_{j+1}$ to get the intermediate result:

$$\beta_{j+1} q_{j+1} = Aq_j - \beta_j q_{j-1} - \alpha_j q_j \qquad (j = 1, 2, \ldots, n-1) \qquad \text{Eqn 2.11}$$

Equations 2.10 and 2.11 are important because they can be used to determine the terms α_j and β_j that define the tridiagonal matrix T, and thus can be used to derive an iteration scheme for generation of new vectors q_{j+1} from the three-term recurrence. If Equation 2.10 is multiplied on the left by q_j^T, Equation 2.11 multiplied on the left by q_{j+1}^T, and the fact that the q_j are orthonormal is taken into account, the remaining terms give these desired definitions:

$$\alpha_j = q_j^T A q_j \qquad \text{Eqn 2.12.a}$$

$$\beta_{j+1} = q_{j+1}^T A q_j \qquad \text{Eqn 2.12.b}$$

Finally, since the q_j are orthogonal, Equation 2.12.a could be rewritten equivalently as:

$$\alpha_j = q_j^T A q_j = q_j^T (Aq_j - \beta_j q_{j-1}) \qquad \text{Eqn 2.13}$$

These relations suggest the following skeletal outline for an algorithm to compute the vector q_{j+1} from the sequence of vectors $q_1, q_2, \ldots, q_j$:

Given $q_1, q_2, ..., q_j$ (and thus α_i and β_i for i = 1 to j, using Eqns 2.12 & 2.13)

Let $r_{j+1} = Aq_j - \alpha_j q_j + \beta_j q_{j-1}$

$\beta_{j+1} = \| r_{j+1} \|$

$q_{j+1} = r_{j+1}/\beta_{j+1}$

$\alpha_{j+1} = q_{j+1}^T A q_{j+1} = q_{j+1}^T (Aq_{j+1} - \beta_{j+1} q_j)$

If $\beta_{j+1} = 0$ at some step of the algorithm, then the iteration must halt or divide by zero. In this case, the tridiagonal matrix is said to be reduced. The interpretation here is that the tridiagonal matrix is locally diagonal, and thus can be decomposed ("reduced") into smaller, independent tridiagonal matrices (one jxj and the other (n-j)x(n-j)). Although this may seem like an unwelcome event, the following discussion demonstrates that it is actually very good news.

Recall that the j^{th} column of AQ = QT was given by $Aq_j = \beta_j q_{j-1} + \alpha_j q_j + \beta_{j+1} q_{j+1}$. If we let $Q_j = [q_1, q_2, ..., q_j]$, and write T_j in terms of α_i and β_i, then

$$AQ_j = Q_j T_j + [0, 0, ..., 0, \beta_{j+1} q_{j+1}] = Q_j T_j + E_j \qquad \text{Eqn. 4.10}$$

where E_j is an nxj matrix whose first j-1 columns are all zero. This type of decomposition of AQ_j is the subject of the following theorem, whose proof (under more general hypotheses) is given in (Golub, 1985).

Theorem 2.3 Let A be an nxn symmetric matrix, T_j be a jxj symmetric tridiagonal matrix, Q_j be an nxj matrix with orthonormal columns, and E_j an nxj matrix that is defined by $AQ_j - Q_j T_j = E_j$. Then the spectrum of T_j approximates that of A, in the sense that there exist j eigenvalues of A $(\lambda_1, \lambda_2, \lambda_3, ..., \lambda_j)$ that satisfy:

$$| \lambda_i - \tau_i | \leq (2)^{1/2} \| E_j \|_2$$

where τ_i (i = 1, 2, ..., j) are the eigenvalues of T_j, and $\| - \|_2$ is the spectral norm.

In the setting of the Lanczos Algorithm, the matrix E_j has a spectral norm given by:

$$\| E_j \| = (\rho(E_j^T E_j))^{1/2} = \beta_{j+1}$$

This means that the vanishing of the off-diagonal term β_{j+1} is equivalent to the calculation of j exact estimates for eigenvalues of A. This "fault" of the algorithm in fact signals its convergence. In practice, none of the off-diagonal terms is ever exactly zero, so the eigenvalues of A are not calculated exactly, but if β_{j+1} ever becomes sufficiently small, the effort to recognize this case and take corrective action is rewarded by a number of nearly exact estimates of A's extremal eigenvalues.

Algorithm 2.2: First Draft, Lanczos Algorithm

Finds: Estimates of extremal eigenpairs of a symmetric matrix A

Given: Initial nonzero vector r_1

(1) Let $\beta_j = \| r_1 \|$ and define $q_0 = 0$

(2) For $j = 1, 2, 3, ..., n - 1$

$q_j = r_j/\beta_j$

$u_j = Aq_j$

$r_{j+1} = u_j - \beta_j q_{j-1}$

$\alpha_j = q_j^T r_{j+1}$

Let $r_{j+1} = r_{j+1} - \alpha_j q_j$

$\beta_{j+1} = \| r_{j+1} \|$

If $(\beta_{j+1} = 0)$ then STOP : the eigenvalues of T_j are exact estimates of j eigenvalues of A.

If $\beta_{j+1} = 0$, so that the algorithm terminates, and it is discovered that more than j eigenvalue estimates are needed, the algorithm can be restarted by choosing a new random initial vector r_1 that is orthogonal to the columns of Q_j. (In practice, exact equality of

floating point numbers is seldom obtained, so the actual test might be $|\beta_{j+1}| < \varepsilon$, where ε is some small error tolerance.)

A serious difficulty plagues Algorithm 2.2, namely the loss of orthogonality among the q_i. Modification of the Lanczos Algorithm to repair this defect requires an analysis of the effect of round-off error on the calculations. This analysis starts with the consideration of the convergence of the algorithm in the absence of round-off. In the following paragraphs, the nxj matrices Q_j and E_j, as well as the jxj matrix T_j, will have the subscript j suppressed, in order to avoid confusion.

Recall the relation $AQ = QT + E$, where $E = [0, 0, ..., 0, \beta_{j+1}q_{j+1}]$ has j columns. Let the eigenpairs of the jxj tridiagonal matrix T be denoted by (θ_i, p_i). Each of these eigenpairs (θ_i, p_i) of T defines an approximate eigenpair (θ_i, v_i) of the matrix A, where v_i is given by $v_i = Qp_i$. The obvious measure of accuracy of the approximation (θ_i, v_i) is the error $\| Av_i - \theta_i v_i \|$ (in the 2-norm), but we would prefer to calculate this error without the expense of multiplying by A. In this vein, we note that:

$$\begin{aligned} \| Av_i - \theta_i v_i \| &= \| AQp_i - \theta_i Qp_i \| && \text{(since } v_i = Qp_i\text{)} \\ &= \| (AQ - QT)p_i \| && \text{(since } \theta_i p_i = Tp_i\text{)} \\ &= \| Ep_i \| && \text{(since } AQ - QT = E\text{)} \end{aligned}$$

If $P = [p_1, p_2, ...,, p_j]$ is the matrix of T's eigenvectors, and the i^{th} column of this matrix is given by the scalars $(\pi_{1i}, \pi_{2i}, ..., \pi_{ji})^T$, then the error $\| Av_i - \theta_i v_i \|$ can be written as:

$$\| Av_i - \theta_i v_i \| = \| Ep_i \| = \| \beta_{j+1}\pi_{ji}q_{j+1} \| = |\beta_{j+1}\pi_{ji}|$$

since all the q_i are of unit length. This last relation gives the desired simple form for the error $\| Av_i - \theta_i v_i \|$: it is just the bottom element of the i^{th} eigenvector of the tridiagonal matrix T, multiplied by the term β_{j+1}. Note that when $\beta_{j+1} = 0$, the error vanishes, in agreement with the results of Theorem 2.3.

In order to use this last result, it is necessary to diagonalize T at each step of the iteration so that the bottom elements of the eigenvectors can be found. This will turn out to be a good idea anyway (from the standpoint of reducing round-off error), but calculating all the eigenvectors of a tridiagonal matrix at every step just to avoid matrix products (for the evaulation of the accuracy of A's eigenpairs) seems like an enormous expense. In fact, the matrix T is only of order j, and j is typically much less than n. In these circumstances, finding the j eigenvectors of T is many times cheaper than performing multiplications by the nxn matrix A.

In practice, after many steps of Algorithm 2.2, it is common to find that, not only are the vectors q_i not orthogonal (i.e, $Q_j^T Q_j \neq I_j$), but that the rank of Q_j is less than j. The vectors q_i are supposed to form an orthonormal set, but in practice may turn out to be linearly dependent. Clearly, something is wrong with the algorithm, and this is the reason that the Lanczos Method was abandoned in the 1950's. In order to understand the reasons for this breakdown of the calculations, the effect of round-off error on the algorithm must be studied. The following discussion is based on that found in (Parlett, 1980b), where the details can be found.

If v_i is an approximate eigenvector of A with associated eigenvalue θ_i, then the analysis of the last section implies that the accuracy of this eigenpair can be measured in terms of the error norm $\| Av_i - \theta_i v_i \| = \| \beta_{j+1} \pi_{ji} \|$, where π_{ji} is the bottom element of the i^{th} eigenvector of $T_j = Q_j^T A Q_j$. Since $v_i = Q_j p_i$, v_i is in the span of $\{q_1, q_2, ..., q_j\}$, so v_i <u>should</u> be orthogonal to the next iterate q_{j+1}, because the q_k are supposed to form an orthonormal set. In practice, however, the inner product $q_{j+1}^T v_i$ may not be zero. In fact, if μ is the unit round-off, then this inner product actually satisfies:

$$| q_{j+1}^T v_i | \leq \mu \| A \| / | \beta_{j+1} \pi_{ji} | = \mu \| A \| / \| Av_i - \theta_i v_i \| \qquad i = 1, 2, ..., j$$

If (θ_i, v_i) has converged to an eigenpair of A, it is possible for the term $\| Av_i - \theta_i v_i \|$ to be arbitrarily small, and thus q_{j+1} can have an arbitrarily large component in the direction of any converged eigenvectors of A (see Figure 2.7 for a geometric interpretation). This is the mechanism by which independence of the iteration vectors q_j is lost. This defect must

be remedied in any practical Lanczos procedure. Two schemes immediately leap to mind:

(1) Orthogonalize q_{j+1} against all the other vectors $q_1, q_2, \ldots, q_j$, and renormalize. In Figure 2.7, this amounts to orthogonalizing q_{j+1} against q_j and q_{j-1}.

(2) Orthogonalize q_{j+1} against all the converged eigenvector estimates $v_i = Q_j p_i$. These estimates can be identified in terms of the error $\| Av_i - \theta_i v_i \| = \| \beta_{j+1} \pi_{ji} \|$. In Figure 2.7, this amounts to orthogonalizing q_{j+1} against v.

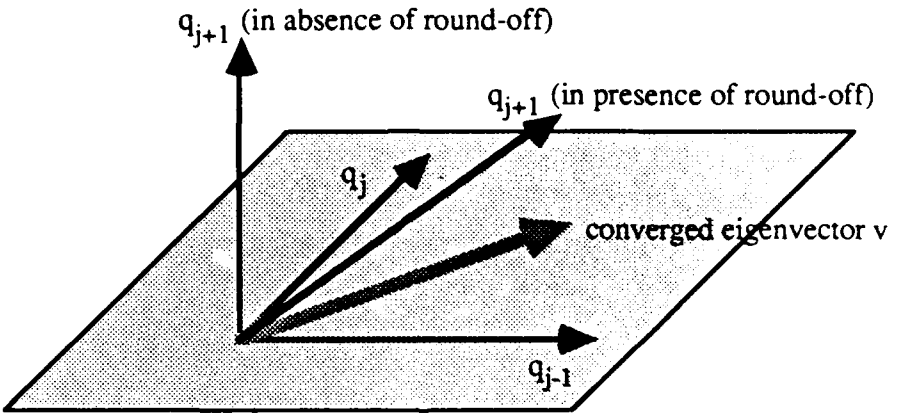


Figure 2.7: Interpretation of Loss of Orthogonality in Lanczos Algorithm

Scheme (1) is termed Lanczos with Complete Reorthogonalization. This remedy was mentioned by Lanczos himself in his original paper (Lanczos, 1950), but it is extremely expensive, and does not directly treat the cause of the problem, namely the converged eigenvectors. Scheme (2) is called Lanczos with Selective Reorthogonalization, and it is expected to be no more expensive than (1), since there cannot be more converged eigenvector estimates than there are vectors q_j. In fact, the expense of finding the bottom elements of T's eigenvectors will increase the cost of scheme (2), but for $j \ll n$, this cost will be insignificant.

The important idea to keep in mind is that the price that must be paid for the control of round-off error in the Lanczos Algorithm is the calculation of the error $\| Av_i - \theta_i v_i \|$ at each step of the iteration. The bad news is that the eigensystem of the jxj tridiagonal matrix

T_j must be found at each step; the good news is that this computation is inexpensive compared to dealing with A, and that this work amounts to checking for convergence of the eigenvector estimates. Such a check for convergence of a desired quantity is a good idea in any iterative scheme. These results allow the following algorithm, which is a robust implementation of the Lanczos Algorithm.

Algorithm 2.3: Lanczos Algorithm with Selective Orthogonalization

Finds: Estimates of extremal eigenpairs of a symmetric matrix A

Given: Initial nonzero vector r_1, error tolerance ε

(1) Let $\beta_j = \| r_1 \|$ and define $q_0 = 0$

(2) For $j = 1, 2, 3, ..., n - 1$

$q_j = r_j/\beta_j$

$u_j = Aq_j$

$r_{j+1} = u_j - \beta_j q_{j-1}$

$\alpha_j = q_j^T r_{j+1}$

Let $r^*_{j+1} = r_{j+1} - \alpha_j q_j$

Let $\beta^*_{j+1} = \| r^*_{j+1} \|$

Compute the eigensystem (θ_i, p_i) of $T_j = \text{tridiag}(\beta_i, \alpha_i, \beta_{i+1})$

For $i = 1, 2, ..., j$

If $| \beta^*_{j+1} \pi_{ji} | < \varepsilon$ then orthogonalize r^*_{j+1} against $Q_j p_i$

Let $r_{j+1} = r^*_{j+1}$

Let $\beta_{j+1} = \| r_{j+1} \|$

If $\beta_{j+1} = 0$ or if sufficient eigenpairs have converged then STOP

As before, if $\beta_{j+1} = 0$, the algorithm can be restarted with an initial vector (typically, a random vector) that is orthogonal to all the q_i, $i = 1, 2, ..., j$.

There are several nice features of this algorithm:

(1) If A has a multiple eigenvalue, then the selective orthogonalization scheme will (eventually) generate an orthonormal set of eigenvectors associated with this eigenvalue. Although the multiplicity itself cannot be guaranteed (there is always the chance that not all the eigenvectors have been found yet), there are schemes (called Block Lanczos) that can determine this multiplicity. The handling of multiple eigenvalues in a robust manner is not a characteristic of very many eigenvalue solvers -- the ability to handle this case is an advantage of the Lanczos scheme.

(2) The algorithm cannot generate an unreduced tridiagonal matrix. Testing for the vanishing of the term $\beta_{j+1} = 0$ prevents an unreduced matrix from appearing. When the QR or QL algorithm is used to diagonalize T_j, this is a very useful feature, since these algorithms only work on unreduced matrices.

(3) The algorithm involves A only in the sense of a matrix product. This means that A need not be stored in explicit form, or even that A exist as an nxn matrix! This last case would include the use of "Inverse Lanczos", in the sense that the Lanczos Algorithm can be applied to A^{-1}, by replacing all the matrix multiplications $y = Ax$ by solutions of the equation $Ay = x$. This case can also include the use of a shift, where the solution is of the equation $(A - \sigma I)y = x$. Although it might seem appropriate to apply the shift in a direct (i.e. non-inverse) setting, by using the matrix multiplication $y = (A - \sigma I)x$, it turns out that the Krylov Subspace is invariant with respect to such a shift, in that $\mathbf{K}(A, u, j) = \mathbf{K}(A - \sigma I, u, j)$. Thus, unlike its cousin the Power Method, no advantage is to be gained by shifting the Lanczos Algorithm, except when the iteration is performed in an inverse setting (since $(A - \sigma I)^{-1} \neq A^{-1} - \sigma I$).

(4) The Algorithm can be easily generalized (Parlett, 1980b) to handle the solution of the matrix pencil problem $(A - \lambda M)u = 0$, where M is a positive-definite matrix. While there are other ways to approach this generalized problem (especially those

involving the Cholesky factorization of M), the Lanczos Algorithm can be extended in a very natural way that retains the other advantages mentioned here.

This type of approach can also be used on the problem of solving linear sets of equations, instead of the solution of the matrix eigenproblem. Depending upon the arrangement of the calculations, either a Lanczos solution procedure or the well-known Conjugate Gradient Method (CGM) is obtained. This topic will be developed further in the next chapter of this document.

Finally, it is worthwhile to reconsider the desired qualities for a "good" approximate solution space CS(Q) that were postulated earlier:

(1) Q^TAQ has to somehow be a "good" approximation to A

(2) Q^TAQ should have some simple structure (i.e. banded, triangular, etc.)

(3) It should be economical to form Q, or to add vectors to Q to increase the rank (and hence the accuracy) of the projected matrix Q^TAQ.

We have seen that the Krylov Subspace is capable of satisfying all three of these criteria:

(1) Q^TAQ is the "best" approximation to A in the sense of the steepest descent optimization of the Rayleigh Quotient.

(2) Q^TAQ is tridiagonal, which is the simplest form a symmetric matrix can possess (except, of course, for a diagonal form, but the diagonalization of an arbitrary symmetric matrix in a finite number of steps violates basic principles of algebra)

(3) The tridiagonal structure of Q^TAQ makes it easy to append new vectors to the solution space using a three-term recurrence.

Overall, it appears that the Krylov subspace is a logical setting for the projection solution of large matrix problems, and that the Lanczos Algorithm is an efficient and robust means to calculate an orthonormal basis for this subspace.

Chapter 3

Development of the Proposed Algorithm

Chapter 3: Development of the Proposed Algorithm

Introduction

The main purpose of the proposed algorithm is to try to reduce the expense of performing a finite element analysis of a large mechanical system. Therefore, much of this chapter is concerned with examining the properties of systems of algebraic equations resulting from Finite Element discretizations. The rest of the chapter is motivated by the desire to construct a reduced coordinate algorithm that will retain the fundamental properties of the larger system, and to place this algorithm in perspective relative to other related methods.

The Finite Element Method is widely used for the generation of approximate solutions of problems in Engineering Mechanics. Generally, a continuum problem involving an infinite-dimensional solution space is reduced to a finite-dimensional matrix problem via a Finite Element discretization. The power of the Finite Element Method lies in its ability to model most of the important properties of the underlying physical problem without introducing severe numerical or implementational difficulties. For instance, if there are material discontinuities or singular applied loads in the physical problem, they can be incorporated without difficulty into the Finite Element model. In a Finite-Difference scheme, this sort of "non-regular" data must generally be simplified, smoothed, or ignored. If the underlying physical problem is governed by a self-adjoint differential relation, the resulting Finite Element equations are typically symmetric (the matrix equivalent of self-adjointness). This sort of preservation of symmetry is not typical of Finite-Difference approximations for the same problem. Finally, when the underlying differential operator is coercive (all of its eigenvalues are positive), the resulting set of Finite Element equations generally involves positive-definite matrices (all of whose eigenvalues are positive). Thus the Finite Element discretization preserves most of the most important physical and mathematical properties of the original physical problem.

Unfortunately, there are a few undesirable characteristics of large physical systems that are preserved in the Finite Element model. The first of these is the sheer size of the problem. Many important physical problems yield Finite Element equation sets that are simply too large to deal with effectively on the computer. A good example of this case is any large three-dimensional problem. Another problem is that the spectrum of the continuum

differential problem contains a wide range of frequencies (these frequencies are often unbounded above). The Finite Element model typically preserves this property in that the frequencies of the discretized problem span many orders of magnitude, so that the matrix problem to be solved may be ill-conditioned. In both of these cases, there is a need for some sort of reduction scheme to take large Finite Element models and simplify them so that:

(1) the important engineering behavior of the solution can be found more inexpensively
(2) the resulting reduced set of equations will be better conditioned, and hence more amenable to a numerical solution.

The development of this chapter is oriented towards the construction of an algorithm to satisfy these two needs, while still retaining the simplicity of implementation that characterizes most Finite Element models. The chapter begins with a general discussion of the Finite Element Method, oriented towards the characteristics of typical Finite Element equations for time-dependent problems. The particular models discussed are dynamic problems, but the methods involved in the construction of the algorithm can be easily used to solve equations whose time-dependence involves diffusive behavior (or even for steady-state problems).

Once the Finite Element discretization has been performed on the spatial terms of a time-dependent problem, the resulting system of temporal differential equations must be solved numerically. This solution process is discussed in the context of one of the most widely-used numerical intergration schemes for dynamic problems, the algorithm known as Newmark's Method. The behavior of the Newmark scheme, as well as the basic principles of time-stepping methods in general, are developed in the next section of this chapter.

The proposed reduced-coordinate algorithm is then developed from the standpoint of a combination of the Lanczos Algorithm and Newmark's Method. Individual components of this proposed algorithm are discussed and suggestions for implementation made. Although the algorithm is cast in terms of Lanczos Vectors and Newmark's Method, the discussion of the proposed algorithm is general enough to permit other projection bases and temporal integration schemes (some of which are mentioned in the last chapter, in connection with suggestions for future research).

Finally, the proposed algorithm is compared and contrasted with other related works from

the recent literature of Computational Mechanics. Since the topic of Projection Methods has a rich and diverse history, only some of the most recent (and most closely related) methods are reviewed in this section.

Note that in this chapter, time is represented by the letter "t", which is traditional for problems of this form. This convention violates those laid out at the end of Chapter 1, since time is a scalar quantity, and scalars are normally written as lower-case Greek letters.

Finite Element Equations

The Finite Element Method is a computational scheme that is commonly used for the numerical solution of the boundary-value problems of mathematical physics. In simplest terms, it is a computationally efficient procedure to interpolate the approximate solution to a physical problem. It is one of the most widely used modelling techniques in engineering, and there are many problems where it is the only rational model that can be used for the generation of numeric solutions. At the heart of the method are two very important concepts:

(1) the construction of a manifold of approximate solutions using locally nonzero interpolation functions.

(2) the application of a Galerkin projection scheme to solve for the optimal member of the solution manifold.

The standard approach to generate a Finite Element Model is to divide the physical body into subregions (elements), incorporating interpolation nodes along element boundaries and within element interiors. Interpolation functions corresponding to each nodal unknown (typically, these unknowns are the nodal displacement components, grouped into a vector) are constructed so that a given interpolation function is equal to unity at its associated node, and equal to zero at every other node. The interpolation functions have narrow support, in that they are nonzero only over elements containing, or contiguous to, the associated node. This narrow support property means that the product of two basis functions is nonzero only when they are both associated with the same element. Therefore, the inner product of two basis functions is "usually" zero, and can be nonzero only when both basis functions are associated with the same element (this condition is referred to as "near-orthogonality").

These interpolation functions form the basis of the approximate solution manifold, and thus serve to define an interpolant for the approximate solution that is globally defined, but depends only on the values of the solution at the nodes (which are of course finite in number).

The Galerkin projection scheme is motivated by the underlying weak physical formulation of the differential problem, e.g., the principle of virtual work. Typically, this weak formulation involves an integral over the physical body, which defines an appropriate inner product function for a Hilbert Space of physical solutions. Seeking the approximate solution in the finite-dimensional manifold amounts to projecting the problem onto the Finite Element basis, which yields a set of algebraic equations in the nodal unknowns. For instance, the integral equations that govern the problem of the dynamic behavior of a deformable continuum give rise to the set of algebraic equations shown in Eqn 3.1:

$$d^T \{ M\ddot{u} + Ku - f \} = 0 \qquad \text{Eqn 3.1}$$

where u is the vector of nodal displacement components,
ü is the vector of nodal acceleration components (often denoted by "a")
f is the vector of applied nodal forces due to external loads
K is the "stiffness" matrix that relates nodal displacement to elastic forces
M is the "mass" matrix that relates nodal accelerations to inertial forces
d is an arbitrary vector of virtual displacements (i.e. a "test" function)

Since we are considering Galerkin schemes, the "test" functions must also lie in the approximate solution manifold, so d has the interpretation of an arbitrary nodal displacement vector. In general, Eqn 3.1 could also include a term Cv, where v is a vector of nodal velocities and C a "damping" matrix. For completeness, this term will be considered in the temporal integration schemes of the next section, but because the product Cv is typically small compared to the other terms in the Eqn 3.1, it will not be included in the following analysis.

For a three-dimensional problem, each nodal displacement consists of three independent displacement components, so if there are N nodes, there will be a total of n = 3N algebraic equations. Since every component of the virtual displacement d is arbitrary, this results in a total of n equations in the n nodal displacement unknowns:

$$M\ddot{u} + Ku = f \qquad \text{Eqn 3.2}$$

It should be noted that K and M will be sparse, because of the near-orthogonality of the basis functions and their derivatives. For any reasonable ordering of the nodes, K and M turn out to be banded, which leads to great savings in computational effort compared to a basis that has global support. In addition, the matrices K and M turn out to be symmetric and positive-definite, corresponding to the self-adjoint and positive character of the underlying integral operators. These properties of the matrices are useful both computationally and theoretically. They make the equation set relatively well-conditioned, guarantee that the point spectrum of both K and M is real, and express the fact that the underlying nature of the continuous integral problem has been inherited by the finite-dimensional discretized form. (In general, none of these characteristics are found in a matrix problem arising from a finite-difference approximation.)

There are two ways to approach the solution of a matrix ordinary differential equation such as Eqn 3.2. The first is to realize that if the mass matrix M is positive definite, then it can be used to define a weighted inner product, which leads to the study of the matrix pencil (K,M). The analytic solution of Eqn 3.2 is then phrased in terms of the (generalized) eigenvalues and eigenvectors of this pencil, and the Lanczos Algorithm is brought into the picture in the generalized sense metioned at the end of Chapter 2. The other approach is to reduce Eqn 3.2 into a matrix problem whose natural inner product is the "usual" n-dimensional space $\mathbf{R}^n$. This approach is a little easier to deal with, since the development of the Lanczos Algorithm in the last chapter was presented in this setting, so the following derivations follow from this standpoint.

If the mass matrix M is positive-definite, then it can be decomposed via the Cholesky Factorization into the product of a lower triangular matrix L and its transpose:

$$M = LL^T$$

In this case, Eqn 3.2 can be multiplied on the left by L^{-1} to obtain the equivalent relation:

$$\ddot{y} + Ay = b \qquad \text{Eqn 3.3}$$

where $\ddot{y} = (\partial^2/\partial t^2)y$

$y = L^T u$

$A = L^{-1} K L^{-T}$

$b = L^{-1} f$ (note that b is a function of time)

Since L is nonsingular whenever M is, the transformations implicit in Eqn 3.3 are invertible, so the coordinates y are derived from the displacement coordinates u by a simple (though nonorthogonal) change of basis. If M is not positive definite, but K is, then the roles of these two matrices can be reversed. If neither M nor K is positive definite, then there is no general theory for the material presented below, since the matrix ODE of Eqn 3.2 cannot be guaranteed to be diagonalizable.

The solution of Eqn 3.3 proceeds by noting that, since A is symmetric (because K is symmetric), the matrix A can be reduced to diagonal form by an orthogonal change of basis:

$$P^T A P = D = \text{diag}(\omega_i^2)$$

where the columns of P are the eigenvectors of A (taken to form an orthonormal set), and the diagonal matrix D consists of A's eigenvalues ω_i^2 (since A is assumed to be positive definite, these eigenvalues are written as squared quantities). The quantities ω_i have the physical interpretation of the natural frequencies of the modes of vibration represented by the eigenvectors of A, or more precisely, by the columns of $Z = L^{-T}P$ (since there is a change of basis involved to obtain Eqn 3.3). Note that the matrix Z is an orthogonal matrix in the inner product weighted by the mass M, since:

$$Z^T M Z = (P^T L^{-1})\, M\, (L^{-T} P) = P^T (L^{-1} L L^T L^{-T})\, P = P^T P = I$$

It is important to realize that there are two Hilbert spaces imposed on this problem. The first is the Hilbert Space $\mathbf{R}^n$, with the inner product $\langle u,v \rangle = u^T v$, and the second is the Hilbert Space characterized by the mass-weighted inner product $\langle u,v \rangle_M = u^T M v$. Recall that the notions of length, orthogonality, convergence, and accuracy are all phrased in terms of the associated inner product, so the "schizophrenia" of these two topological settings

should be kept in mind at all times. What is also important (and comforting) to remember is that the time scales of the problem are dictated by the frequencies ω_i, which are the same for both problems.

Although the computational schemes presented in this work do not attempt to diagonalize exactly the matrix problem defined by Eqn 3.2, it is instructive to consider the behavior of this diagonal problem for the linear case, since it leads to some insights that should be considered in the proposed algorithm and its relatives. Therefore, an overview of the theory for the analytic solution in this case will be presented in the following paragraphs.

The change of basis induced by the matrix P on the problem of Eqn 3.3 diagonalizes the matrix system of ODE into a set of scalar ODE:

$$\partial^2\psi_i/\partial t^2 + \omega_i^2 \psi_i = \phi(t)\, \varphi_i \qquad \text{Eqn 3.4}$$

where ψ_i is the contribution of the i^{th} eigenvector to the solution (if $z = (\psi_1, \psi_2, ..., \psi_n)^T$, then the vector y is given by $y = Pz$), and the term $\phi(t)\, \varphi_i$ is equal to $p_i^T b$, where p_i is the i^{th} column of P. The time-dependence of this modal loading term is included in the first factor for emphasis that the load b can be a function of time. The solution to Eqn 3.4 will be a sum of a homogenous solution (trigonometric functions with frequency ω_i), and a particular solution that depends upon the right-hand side. In particular:

(1) If p_i has a large component in the direction of the spatial distribution of b, then the i^{th} mode is said to participate in the solution, and this mode may have the potential to contribute significantly to the solution of Eqn 3.2.

(2) If $\phi(t)$ contains a large frequency component that is close to ω_i, then the i^{th} mode may also be expected to contribute significantly to the solution of Eqn 3.2, through the phenomenon of resonance.

In either case, the overall behavior of the solution of Eqn 3.2 may be largely influenced by a particular modal contribution due to a spatial or temporal matching of the loading terms to

the natural vibrational modes of the structure. Any reduced coordinate model that will be expected to capture the important mechanical behavior of the larger problem must incorporate some means of including these effects.

It should be mentioned that the lower frequencies of the discretized problem given by Eqn 3.2 tend to be much more accurate estimates of the actual frequencies of the underlying continuous problem than the higher ones. As an example, many continuous problems have a point spectrum that is unbounded, yet there is no way that a finite-dimensional operator can exhibit this sort of behavior. It will be seen in the next section that these higher frequencies can cause serious difficulties for the unreduced problem. This is not such a surprising result, since the spread of these time scales is obviously related to the condition of the problem.

Finally, we note that the diagonalization used to examine the matrix problem is not strictly applicable in a nonlinear setting, since then the matrices, the associated frequencies, and the modes defined by the columns of P (or Z) all evolve with time. (In fact, it may be somewhat of a misnomer to refer to the eigenvalues in terms of frequencies, since the interpretation of a frequency as representative of a characteristic period for a mode to return to its initial state may not be appropriate in a nonlinear problem). Nonetheless, the qualitative ideas presented are still important in a nonlinear problem (especially those that pertain to contributions due to spatial and temporal matching of load to response), and we shall see in the next section that the practical solution of the time-dependence of the solution depends upon a iteration scheme that involves the solution of a linear system of equations at each step.

Temporal Integration Schemes

The equations of motion form an n-dimensional set, but there are 3n unknowns at each time (displacements, velocities and accelerations). Some method of reducing the size of this problem must be employed in order to achieve a unique solution at the end of the time step. Newmark's Method (Newmark, 1959) postulates simple polynomial relations among these coordinates:

$$v_{n+1} = v_n + (1 - \gamma)a_n h + \gamma a_{n+1} h \qquad \text{Eqn 3.5.a}$$

$$u_{n+1} = u_n + v_n h + (1/2 - \beta)a_n h^2 + \beta a_{n+1} h^2 \qquad \text{Eqn 3.5.b}$$

where the length of the time step is given by $h = t_{n+1} - t_n$ (note that h is a scalar), and a subscript "n" represents the value of the corresponding quantity at time t_n (i.e., $u_n = u(t_n)$).

These equations can be taken as a simple expression of polynomial relations among the displacement u, velocity v, and acceleration a, or they can be derived by a weak formulation of the temporal problem, with the "test" functions in the form of a two-parameter family involving γ and β. The details of this latter approach can be found in (Zienkiewicz, 1977, Chapter 21), along with the generalization to families of numerical integration schemes that involve three or more parameters.

This type of polynomial approximation has been used very successfully in a wide variety of problems in structural dynamics. Newmark's original derivation for the two-parameter numerical integration scheme based on these relations used the accelerations as the solution of the equations of motion at time t_{n+1}. The velocities and displacements can then be obtained by substitution of the quantity a_{n+1} into Eqn 3.5. Alternatively, the relations of Eqn 3.5 can be recast so that the displacements at time t_{n+1} are the primary unknowns, and the velocities and accelerations become derived quantities. Some of the details of the development of these schemes will be presented later in this section.

In any numerical method, the primary issue that must be addressed is that of convergence. A convergent method is one that guarantees that a refinement of the discretization will produce generally more accurate results. In a time-stepping scheme, as the size of the time step decreases, the answers converge to the correct solution. A second criterion is that of accuracy, which is related to that of convergence. Where convergence addresses the question "does the error go to zero as the time step decreases?", accuracy is concerned with "at any particular time step, how accurate is the solution?", or perhaps "at what rate does the error go to zero with step size?". Obviously, a convergent method in which the error goes to zero as the square of the step size (a quadratic convergence rate) will eventually become more accurate than another scheme in which the error and step size decrease at the same rate (a linear convergence rate).

Another concern is that of stability — a stable method is one that guarantees that errors introduced at one step cannot grow with successive steps. If a method is unstable, even when the errors at each step are small, they can increase exponentially with time, thus overwhelming the solution. In a stable method this cannot occur, although stability alone does not tell us anything about the size of solution errors that may be introduced at each step, or whether errors from different times can grow by accumulation. Many numerical methods are only conditionally stable, in that stability is guaranteed only when the step size is smaller than some threshold time scale dictated by the data of the problem and the discretization (typically, this time scale is a factor of the shortest period of vibration for the structure). Some idea of this critical time scale must be known a priori for a conditionally stable method to behave in a robust manner. For this reason, the use of unconditionally stable methods is often preferred — these methods are stable regardless of the step size (although the actual size of the errors introduced at each step may still be large). Finally, it should be noted that there are many related definitions of stability, and a precise definition is a matter of opinion in the nonlinear case. The reader is referred to (Hughes, 1983) for an exhaustive and yet very readable view of this topic. (Much of the detail underlying this section can be found in this reference.)

The convergence and stability characteristics of a numerical method are not independent (they are related by the Lax Equivalence Theorem, which is often termed "The Fundamental Theorem of Numerical Analysis"), and so it is no surprise that we would want to restrict ourselves first to methods that are both convergent and at least conditionally stable. In addition, it should be clear that an unconditionally stable method, especially if it has a higher order convergence rate, is to be desired. In the context of Newmark's Method, it can be shown (Hughes, 1983) that the algorithm derived from the relations of Eqn. 3.5 will be:

(1) unconditionally stable when $2\beta \geq \gamma \geq 1/2$
(2) linearly convergent when $\gamma \neq 1/2$
(2) quadratically convergent when $\gamma = 1/2$

An obvious (and widely used) choice for these parameters is $\beta = 1/4$ and $\gamma = 1/2$. In this case, the time-stepping strategy has the particularly simple interpretation of applying the trapezoidal rule to the integration of $a = dv/dt$, $v = dx/dt$:

$$v(t_{n+1}) = v(t_n) + \int_{t_n}^{t_{n+1}} a(t+\tau)\, d\tau \qquad \text{Eqn 3.6.a}$$

$$= v(t_n) + (h/2)[a(t_{n+1}) + a(t_n)]$$

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} v(t+\tau)\, d\tau \qquad \text{Eqn 3.6.b}$$

$$= u(t_n) + (h/2)[v(t_{n+1}) + v(t_n)]$$

$$= u(t_n) + hv(t_n) + (h^2/4)[a(t_{n+1}) + a(t_n)]$$

This choice is often termed the trapezoidal method, or alternatively, the average-acceleration method.

Newmark proposed a simple but effective iteration scheme for nonlinear problems. The equations of motion are solved for the acceleration at the end of the time step (recall Eqn 3.2, evaluated at time t_{n+1}), and then this new acceleration estimate can be used to find improved values of velocities and displacements.

$$^{(i+1)}a_{n+1} = M^{-1}(f_{n+1} - K\,{}^{(i)}u_{n+1} - C\,{}^{(i)}v_{n+1}) \qquad \text{Eqn 3.7}$$

where the superscript in parentheses preceding a quantity refers to the iteration number. Note that the effects of damping have been reincorporated into these equations.

To examine the convergence of this iteration for a particular time step, the iteration can be written in the form:

$$^{(i+1)}a_{n+1} = F(\,{}^{(i)}a_{n+1}) + b \qquad \text{Eqn 3.8}$$

Where the vector b contains all the terms that do not depend on the latest estimate of the acceleration. The convergence of this iteration is related to the existence of attractive fixed points for the transformation F. The particular form of the transformation matrix F can be evaluated by substitution of the governing relations among displacement, velocity, and acceleration (given by Eqn 3.5) into the iteration defined by Eqn 3.7:

$$
\begin{aligned}
{}^{(i+1)}a_{n+1} &= M^{-1}\,(p - K\,{}^{(i)}u_{n+1} - C\,{}^{(i)}v_{n+1}) \\
&= M^{-1}\,(p - K\,\{u_n + v_n h + (1/2 - \beta)a_n h^2\} - C\,\{v_n + (1 - \gamma)a_n h\} \\
&\quad - \beta h^2 K\,{}^{(i)}a_{n+1} - \gamma h C\,{}^{(i)}a_{n+1}) \\
&= - hM^{-1}(\beta hK + \gamma C)\,{}^{(i)}a_{n+1} + (\text{terms independent of } a_{n+1}) \\
&= F\,{}^{(i)}a_{n+1} + b
\end{aligned}
$$

In this case the transformation operator F is the matrix defined by:

$$F = - hM^{-1}(\beta hK + \gamma C) \qquad \text{Eqn 3.9}$$

A sufficient condition for the convergence of this iteration is that the solution a_{n+1} behave as a fixed point of the transformation F:

$$a_{n+1} = F\,a_{n+1} + b \qquad \text{Eqn 3.10}$$

In the linear case Eqn 3.10 can be subtracted from Eqn 3.8 to obtain estimates of the error ${}^{(i)}e_{n+1}$ at the i^{th} iteration:

$$
\begin{aligned}
{}^{(i+1)}e_{n+1} &= {}^{(i+1)}a_{n+1} - a_{n+1} \\
&= F\,{}^{(i)}a_{n+1} + b - (F\,a_{n+1} + b) \\
&= F\,({}^{(i)}a_{n+1} - a_{n+1}) \\
&= {}^{(i)}e_{n+1}
\end{aligned}
$$

$$\therefore \quad {}^{(i+1)}e_{n+1} = F\,{}^{(i)}e_{n+1} \qquad \text{Eqn 3.11}$$

Eqn 3.11 implies that the convergence of the Newmark iteration defined by Eqn 3.7 is guaranteed whenever the spectral radius of the transformation F is less than unity. The following development shows that this will occur in the undamped case when the time step is smaller than a multiple of the smallest characteristic period of the pencil (K, M). This characteristic period is given by:

$$T_j = 2\pi/\omega_j$$

where the scalar ω_j^2 is the largest eigenvalue of the pencil (K, M). This result is obtained by using the fact that the eigenvalues of $M^{-1}K$ are the same as those of the symmetric matrix $A = L^{-1}KL^{-T}$, where M has the Cholesky factorization $M = LL^T$. In this case, we find that:

$$\rho(F) = \rho(\beta h^2 M^{-1}K)$$

$$= | \beta h^2 | \rho(M^{-1}K)$$

$$= | \beta | h^2 \omega_j^2$$

$$= | \beta | (2\pi)^2 (h/T_j)^2 < 1 \quad \text{for convergence}$$

Therefore, a sufficient condition for the Newmark iteration to converge is that the step size h should be taken smaller than the smallest period multiplied by the factor $1/(2\pi \beta^{1/2})$.

This result is not good news: the point of the trapezoidal version of the Newmark Method is that it gives an unconditionally stable algorithm, so that the length of a time step is not dictated by stability conditions that restrict the step size to scales on the order of the shortest vibrational period of the structure. Unfortunately, the convergence of the iteration used to find the desired accelerations gives exactly these same conditions that we are trying to avoid by using an unconditionally stable method. There is hope that these conditions, though sufficient, are not necessary, so that they may overly conservative in helping us choose a step size. For a linear system, the calculations could conceivably be reordered so that no iteration is necessary, but this sort of detail will not help in the nonlinear case, since iteration will be required whenever the mass, stiffness. or load depend upon the solution.

At this point it is instructive to recall that the highest frequencies of vibration for the structure (which cause stability and convergence concerns) are usually not very accurate estimates of the associated modes and frequencies of the actual physical system being modelled. Some thought might be given to the idea of "filtering out" these high-frequency effects, both because they cause stability or convergence trouble and also because they arise from inaccuracies introduced by the discretization. One way to perform this filtering would be to incorporate artificial damping into the damping matrix C. Unfortunately, this has the effect of filtering out the modes corresponding to the midrange of frequencies, and leaving

the higher modes intact (see Hughes, 1983). Another way to perform this filtering is to vary the parameters β and γ in order to introduce some artificial dissipation of energy in the higher modes. Again, this approach causes some difficulties, including the concern that the high rate of convergence may be lost when $\gamma \neq 1/2$. Still another method is to abandon the Newmark algorithm altogether, and use another method, even at the expense of more implementation difficulties (recall that there are families of methods with more available parameters than the two given by the Newmark family).

There is another approach which should be mentioned. If the problem defined by Eqn 3.2 is projected onto a subspace associated with the lower modes of vibration (or some approximation of this subspace), the resulting projected problem does not contain the high-frequency behavior of the full (unreduced) equation that is causing these difficulties. Care will have to be taken not to lose any information that may be particularly important in such a case (e.g., resonance effects for midrange frequencies), but the general idea of filtering high frequencies via some sort of a projection scheme appears to have promise.

An alternative formulation of the Newmark scheme can be derived that uses displacements at the end of the time step as the primary unknowns, instead of accelerations. Consideration of this form of the algorithm begins by rewriting the relations of Eqn 3.5 so that velocity and acceleration at the end of the time step are expressed in terms of the increment of displacement:

$$v_{n+1} = (\gamma/\beta h)(u_{n+1} - u_n) + (1 - \gamma/\beta)v_n + (1 - \gamma/2\beta)a_n h \qquad \text{Eqn 3.12.a}$$

$$a_{n+1} = (1/\beta h^2)(u_{n+1} - u_n) - (1/\beta h)v_n + (1 - 1/2\beta)a_n \qquad \text{Eqn 3.12.b}$$

Now the equilibrium equations at the end of the time step ($t = t_{n+1}$) can be written entirely in terms of the displacement vector at this time:

$$K_{eff}\, u_{n+1} = f_{eff} \qquad \text{Eqn 3.13}$$

where $K_{eff} = K + (\gamma/\beta h)\, C + (1/\beta h^2)\, M$

$$f_{eff} = f_{n+1} + C\,[(\gamma/\beta h)u_n - (1 - \gamma/\beta)v_n - (1 - \gamma/2\beta)a_n h]$$

$$+ M\,[(1/\beta h^2)u_n + (1/\beta h)v_n - (1 - 1/2\beta)a_n$$

After Eqn 3.13 is solved for u_{n+1}, the relations of Eqn 3.12 can be used to find estimates for the velocity v_{n+1} and acceleration a_{n+1}. In general, these state variables (u, v, and a) will not satisfy the equilibrium equations at the end of the time step, and so some form of iteration can be constructed. In particular, this form of Newmark's method admits a Newton-like iteration scheme, whenever a "tangent stiffness" K_t can be evaluated. In the following derivation, it will be assumed that the nonlinearities of the problem are expressed so that only K and f depend on the displacement u. This implies that the mass and damping matrices are not functions of the state variables, and that K and f do not depend on velocities or acceleration (this situation includes both the consideration of small-deformation plasticity and simple nonlinear boundary conditions). Relaxation of these conditions is not difficult, but the nomenclature gets a little more complicated, so the general case will not be treated here.

Define a residual vector r(u) by the relation:

$$r(u) = K(u)u + Cv + Ma - f(u) \qquad \text{Eqn 3.14}$$

When consideration is made for the fact that v and a can be expressed as simple functions of displacement u (recall Eqn 3.12), then Eqn 3.14 is seen to be a nonlinear set of n equations in the n displacement components. The derivative of r with respect to u is given by:

$$r'(u) = \{(\partial K/\partial u)u + K\} + C(\partial v/\partial u) + M(\partial a/\partial u) - \partial f/\partial u$$

or

$$r'(u) = \{(\partial K/\partial u)u + K\} + (\gamma/\beta h)C + (1/\beta h^2)M - \partial f/\partial u \qquad \text{Eqn 3.15}$$

where the derivatives of velocity and acceleration with respect to displacement have been obtained from the defining relations of Eqn 3.12. The term in braces is the tangent stiffness matrix, which may either be exact (assuming the third-rank tensor $\partial K/\partial u$ can be evaluated) or approximate, as is often the case in plasticity problems.

Since Eqn 3.14 defines a set of nonlinear equations, and Eqn 3.15 shows how to evaluate the gradient of these equations, these two relations can be combined into an algorithm for using Newton's method to solve the nonlinear equilibrium equations at the end of the time step:

$$\text{Solve } r'({}^{(i-1)}u)\,[{}^{(i)}u - {}^{(i-1)}u] = -\,r({}^{(i-1)}u) \qquad \text{for } i = 1, 2, \ldots \text{ until converged}$$

(The subscript "n+1" has been suppressed for clarity.)

As usual with Newton schemes, some starting vector must be obtained — for this purpose, a rough estimate of the displacement ${}^{(o)}u_{n+1}$ (such as the displacement at the end of the last time step) is used in conjunction with Eqn 3.13 to obtain a prediction for ${}^{(1)}u_{n+1}$. This prediction is supplemented by the correction obtained by using Newton's method to find ${}^{(i)}u_{n+1}$ for iterations i = 2, 3, ... until convergence is obtained.

Algorithm 3.1: Newmark-Newton Temporal Integration

Finds: displacement, velocity, and acceleration at time t_{n+1}

Given: an initial estimate of the displacement ${}^{(o)}u_{n+1}$

(1) Initialization: evaluate K, f, C, and M using the estimate ${}^{(o)}u_{n+1}$

calculate ${}^{(o)}v_{n+1}$ and ${}^{(o)}a_{n+1}$ using Eqn 3.12

(2) Predictor: use Eqn 3.13 to obtain the estimate ${}^{(1)}u_{n+1}$

use Eqn 3.12 to obtain ${}^{(1)}v_{n+1}$ and ${}^{(1)}a_{n+1}$

(3) Corrector: For i = 2, 3, ... until converged:

solve $r'({}^{(i-1)}u_{n+1})\,[{}^{(i)}u_{n+1} - {}^{(i-1)}u_{n+1}] = -\,r({}^{(i-1)}u_{n+1})$

calculate ${}^{(i)}v_{n+1}$ and ${}^{(i)}a_{n+1}$ using Eqn 3.12

The convergence of this iteration scheme can be examined using the same sort of fixed-point analysis that was derived for the original Newmark scheme earlier in this section. The iteration amounts to the transformation of ${}^{(i)}u_{n+1}$ by a tranformation matrix F, whose spectral radius must be less than one for the iteration to have an attractive fixed point. In the following analysis, in order to call attention to the important result, it is assumed that there is no damping and that the load f is independent of the displacement u (this allows the

iteration to be phrased in terms of the natural frequencies of the discretized structure).

Finally, it is assumed that the time step h is sufficiently small so that $(1/\beta h^2)M$ dominates the stiffness K in the effective stiffness matrix — this assumption admits an asymptotic approximation for the inverse of r'(u).

Begin the analysis by writing the iteration in the usual transformation form:

$$^{(i+1)}u_{n+1} = F(\,^{(i)}u_{n+1}) + b \qquad \text{Eqn 3.16}$$

Where the vector b contains all the terms that do not depend on the latest estimate of the displacement. Substitution of the governing relations among displacement, velocity, and acceleration (given by Eqn 3.12) into the iteration defined by Eqn 3.16 yields:

$$\begin{aligned}
^{(i+1)}u_{n+1} &= {}^{(i)}u_{n+1} - [r'(({}^{(i)}u_{n+1})]^{-1}\,[r(({}^{(i)}u_{n+1})] \\
&= {}^{(i)}u_{n+1} - [K_t + (1/\beta h^2)M]^{-1}\,[\{K + (1/\beta h^2)M\}\,{}^{(i)}u_{n+1} - f] \\
&= {}^{(i)}u_{n+1} - [(\beta h^2)M^{-1} + (\beta h^2)^2 M^{-2}K_t + ...]\,[\{K + (1/\beta h^2)M\}\,{}^{(i)}u_{n+1} - f] \\
&= {}^{(i)}u_{n+1} - [(\beta h^2)M^{-1}\,(1/\beta h^2)M]\,{}^{(i)}u_{n+1} + [(\beta h^2)M^{-1}K]\,{}^{(i)}u_{n+1} + ... \\
&= {}^{(i)}u_{n+1} - {}^{(i)}u_{n+1} + [(\beta h^2)M^{-1}K]\,{}^{(i)}u_{n+1} + ... \\
&= [(\beta h^2)M^{-1}K]\,{}^{(i)}u_{n+1} + ... \qquad \text{Eqn 3.17}
\end{aligned}$$

The ellipsis (...) indicates higher-order terms in $^{(i)}u_{n+1}$, as well as terms that are independent of this displacement vector.

Eqn 3.17 is a similar result to that derived earlier for the original Newmark iteration. As before, a sufficient condition for the error to tend to zero as the iteration continues is that the time step h satisfy the implicit condition:

$$|\beta|\,(2\pi)^2\,(h/T_j)^2 < 1 \quad \text{for convergence}$$

This convergence condition for the Newmark-Newton iteration defined by Algorithm 3.1 to converge is that the step size h should be taken smaller than the smallest period multiplied

by the factor $1/(2\pi\beta^{1/2})$. As we saw earlier, this is the same step-size limitation that is being avoided by the use of unconditionally stable schemes (such as this one). Thus, although the sophistication of Newton's Method is being employed for the iteration scheme, the sufficient condition for the convergence of this iteration still involves the highest frequencies of the structure. Clearly, the idea of a projection as a filter to remove these troublesome high frequencies is worthy of closer attention.

Finally, anyone who has ever used a temporal integration scheme knows that the subdivision of the time step usually gives better results, both in terms of accuracy (since the error in replacing temporal derivatives with differencing schemes gets smaller with the time step), and in terms of conditioning (with the numerical behavior of the matrix equation used to integrate the solution improving as the time step is shortened). There is thus a good reason for any temporal integration scheme to use a sub-incrementing procedure whenever convergence to the correct solution at the end of the time step cannot be obtained. It is worthwhile to examine this fact from the standpoint of the convergence of the Newmark-Newton iteration given in Algorithm 3.1. In order to simplify the analysis, the matrix pencil (K, M) used in Eqn 3.13 will be replaced by the equivalent problem (A, I), so that the underlying matrix relation to be solved at each step is:

$$A_{eff}\, u = f_{eff} \qquad \text{Eqn 3.18}$$

where A_{eff} is the effective stiffness obtained by replacing M with I and setting C to zero in the defining relations for Eqn 3.13. Then the matrix problem to be solved at each step takes the form:

$$(A - \lambda I)u = f_{eff} \qquad \text{with} \quad \lambda = -(1/\beta h^2)$$

This form allows the use of the Fredholm Alternative to examine the condition of A_{eff}. Since A's eigenvalues (the frequencies ω_i^2) are all positive, the parameter λ cannot be an eigenvalue of A. Application of the Fredholm Alternative Theorem implies that the solution of Eqn 3.18 will have components in the direction of the i^{th} eigenvector of A whose magnitude varies inversely with $(\omega_i^2 + 1/\beta h^2)$. Thus, the condition of A_{eff} looks like:

$$\rho(A_{eff}) = (\omega_n^2 + 1/\beta h^2)/(\omega_1^2 + 1/\beta h^2)$$

This can be compared to the static case (M = 0), where the condition of A is given by:

$$\rho(A) = \omega_n^2/\omega_1^2$$

Obviously, as h decreases, the condition of the matrix problem for the solution at the end of the time step improves (the condition number goes to unity). This results confirms the well-known qualitative principle that a dynamic problem involving a nonlinear stiffness matrix K will be "better-behaved" than a nonlinear static problem involving the same matrix. This also gives a good reason to consider sub-stepping in the time domain whenever the nonlinear iteration for the solution at the end of the time step will not converge.

The Reduced Coordinate Algorithm

The basic principles of the proposed reduced coordinate algorithm have been developed in various sections of this document, and they now can be assembled here. The most important components of the method are:

(1) Generation of an appropriate manifold for the projection solution of the set of Finite Element equations using the Krylov Subspace given by the Lanczos Algorithm in an inverse setting. The initial vector for the Lanczos Algorithm may include the spatial variation of the forcing function. If resonance is expected, a shift may be used to generate Lanczos vectors that are the most likely to be excited by the time-dependent forcing function. Either the Lanczos vectors themselves (spanning the Krylov Subspace) or a number of converged eigenvectors from the Lanczos Algorithm may be used. The former case will give basis vectors that will be termed "Lanczos vectors" and the latter will yield "Lanczos eigenvectors".

(2) Use of an appropriate temporal integration scheme for the integration of the reduced equations of motion. "Appropriate" in this sense means that the primary unknowns of the time-stepping scheme (which will be forced to lie in the approximation subspace) may need to be chosen so as to match the physical interpretation of the vectors used to construct the approximation manifold. If the solution exhibits enough differentiability, the notion of "appropriate" in this sense may be relaxed somewhat.

(3) Incorporation of a "sub-stepping" scheme into the temporal integration scheme to enhance the robustness of the algorithm. If this sub-stepping does not yield the desired convergence of the equilibrium equations at the end of the time step, then the algorithm should either halt, or attempt to generate a new approximation manifold using the last accepted values of the state variables u, v, and a.

The purpose of this section is to use these principles to construct a relatively coherent form of the proposed algorithm.

The matrix equation to be solved is the dynamic equilibrium relation of Eqn 3.2:

$$M\ddot{u} + Ku = f$$

where K and f can be functions of the displacement vector u. In general, this relation could include a damping matrix C and dependence of all the terms on the displacement and its derivatives. (These extensions are easily made, but because they are not required at the present stage of the research, they will not be considered here.) This equation can be projected onto an approximating subspace spanned by the orthonormal (with respect to the mass-weighted inner product) columns of Q to obtain the reduced set:

$$\ddot{y} + Ry = g \qquad \text{Eqn 3.18}$$

This matrix equation can now be integrated over time to give the desired solution $u = Qy$. It should be emphasized that this temporal integration is equivalent to the solution of the nonlinear equilibrium equations at the time step $t = t_n$, using the solution at the last time step as initial conditions. Since these equations are generally nonlinear, some type of iteration scheme will be used, such as the original Newmark Method, or the Newton scheme of Algorithm 3.1. If the iteration converges, then the next time step is considered. If the iteration for the solution at time t_n does not converge, there are a few actions that can be taken:

(1) Subdivide the time step to get a more well-conditioned problem. This gives a temporal integration scheme with sub-stepping, and as long as the time step is broken into some integral number of substeps, the solution will eventually be obtained at time t_n (assuming that the iteration converges for each subincrement).

(2) Recalulate a new projection basis. If the effect of the nonlinearities is to alter the matching in space or time between the forcing terms and the approximation manifold, a new manifold will have to be constructed. An example of this would occur when some elements of the Finite Element mesh undergo large plastic deformations, so that the original approximation space is no longer a good estimate of the shape of the deformed structure. In this case, care must be taken to make the solution continuous as the manifold is modified, so that the "jump" from one solution manifold to another does not introduce a discontinuous solution in time. One way to achieve this continuity is to consider the projection solution of the problem

$$M(\ddot{u} - \ddot{u}_o) + K(u - u_o) = f$$

where u_o is the last approximate solution calculated before the manifold was recalculated. If this approach is used to define a new incremental displacement vector $z = u - u_o$, then the projection solution z(t) will be a continuous function of time.

(3) If neither of these schemes gives a convergent iteration for the solution, then the algorithm will have to be halted. This is always a consideration that must be dealt with in a nonlinear solution method, and alternatives to "giving up" when choices (1) and (2) do not give satisfactory results are being considered by the authors.

Implicit in the preceding discussion is the choice of the approximation subspace — depending upon this choice, one of many related algorithms can be constructed. The approach used in this research is to generate the approximation manifold using the Lanczos Algorithm in an inverse setting. This inverse approach is motivated by the spectral representations for the solution given by the Fredholm Alternative, which showed that the minimal end of the spectrum is the most important for approximation purposes. Unfortunately, the generation of Lanczos vectors in an inverse setting involves solving the large system of equations that we are trying to avoid by using a reduced coordinate scheme. Because these vectors are so expensive to generate, it is desirable to reuse this approximation space over many time steps, so that the cost of finding the Lanczos vectors can be amortized over these many steps.

These considerations allow the statement of a relatively concrete version of the proposed reduced coordinate algorithm. Some of the details of this algorithm will be discussed in the next section, and relatives of this algorithm obtained by modifications of some of the basic principles discussed above will be given in the final section of this chapter.

Algorithm 3.2: Reduced Coordinate Algorithm

Finds: displacement history for the problem $M\ddot{u} + Ku = f$

Given: initial conditions on displacement and velocity, error tolerance ε, length of time for analysis t_{max}, time steps h, maximum number of temporal iterations ITMAX

(1) Initialization: evaluate initial modes of problem and arrange them as the columns of Q

(2) For $t = t_1$ to t_{max}

(2a) For i = 1 to ITMAX

solve $^{(i)}\ddot{y} + R\,^{(i)}y = g$ for estimate of reduced solution $y(t_n)$

evaluate residual $r = {}^{(i)}\ddot{y} + R\,{}^{(i)}y - g$

if $\| r \| < \varepsilon$, then
iteration has converged
accept estimate y_n for solution at time $t = t_n$
break to next time step

else if $i <$ ITMAX
continue iteration

else
iteration has not converged in ITMAX steps

if time step has not been subincremented then
subincrement time step h and continue from (2)

else if projection basis Q has not been updated
calculate new basis Q and continue from (2a)

else
terminate algorithm due to lack of convergence

Computational Considerations

The algorithm of the last section is still somewhat nebulous, so this section is devoted to clarifying some important basic ideas. The most important questions include "how expensive is this algorithm?", "what should the dimension of the approximating subspace be?", and "what can go wrong with this scheme?". These questions will be addressed in that order.

The most expensive part of almost any Finite-Element analysis is the formation and solution of the governing matrix equation. The cost of forming the element matrices can be very difficult to quantify, but in many nonlinear problems it may be the most expensive step. For an unreduced problem, the cost of performing one iteration of a time-stepping scheme can be decomposed into the cost of forming the matrix equation from element contributions and of solving the resultant set. The element cost can be represented by:

NELEM*FORMCOST

where NELEM is the number of elements and FORMCOST is an estimate for the average cost of forming an element matrix. The cost of solution of the system of equations increases like:

$NEQ*NBAND^2$

where NEQ is the number of equations and NBAND is the bandwidth (in some average sense, if a "profile" solver is used). There may be other costs associated with a Finite-Element analysis, but these two types are typically the largest fraction of the computational expense. In this case the total cost of the solution can be estimated by:

$$COST = \alpha*NELEM*FORMCOST + \beta*NEQ*NBAND^2 \qquad \text{Eqn 3.19}$$

for some scalars α and β.

In a reduced coordinate algorithm, the system of equations can be formed by performing the reduction (pre- and post-multiplication by Q or Q^T) at the global or at the element level. Usually, this calculation would be done at the element level to save storage space, so the formation of the equations would involve computational effort that grows like:

$$NELEM*FORMCOST + \gamma*NMOD^2$$

where NMOD is the number of modes in the reduced formulation. The parameter γ depends on the size of the element matrices, and this size helps to determine whether the formation of the element arrays or their transformation into reduced form constitutes the larger cost. Once the reduced arrays are found, the solution costs for the reduced model at each iteration step grow like:

$$NMOD^3$$

So the total cost of the reduced model for one iteration can be estimated to be:

$$\alpha*NELEM*FORMCOST + \delta NMOD^2 + \varepsilon*NMOD^3 \qquad \text{Eqn 3.20}$$

Although it is difficult to compare Eqns 3.19 and 3.20 since the coefficients of each term are unknown, there are a few conclusions that can be drawn:

(1) The cost of forming the element matrices represents a computational overhead that must be performed for either method. If this cost constitutes the major fraction of the effort required, then the two algorithms will be competitive.

(2) The cost of solving the system of equations is very different for the two methods. Whenever the bandwidth and the number of equations grow simultaneously (i.e. in all but one-dimensional problems), the cost of solving the reduced equations has the potential to be much cheaper that that associated with an unreduced problem. In particular, for large two-dimensional and almost all three-dimensional problems, unless an inordinately large number of modes are required for the solution, the reduced algorithm will be cheaper to implement at each iteration.

The next important question to be addressed involves the size of the approximating subspace required for good accuracy. This question is very difficult to answer in general terms, because a small residual error in the reduced problem may correspond to a large error in the actual unreduced problem. In turn, this error in the large discretized model may represent an even larger mistake in the setting of the underlying continuous problem. Knowing how many modes to include in a solution is a little like knowing (a priori) how many elements will be sufficient to gain a given accuracy in an arbitrary Finite-Element

discretization. Because the problem is satisfied only in a projected sense, it is not clear what the actual (unprojected) error is. Nonetheless, there are a few suggestions that can be made:

(1) If the forcing function can be represented by a spatial term f (such as in an earthquake problem, where this vector is the product of the mass matrix and a vector of influence coefficients) the norm of the projection error $(I - QQ^T)f$ can be monitored. Once this error decreases to some preset tolerance (e.g. 5%), it is clear that enough modes have been found so as to model the spatial variation in the loading function. (Recall that $(I - QQ^T)f$ the component of f that is orthogonal to the columns of Q in the "usual" inner-product for $\mathbf{R}^n$.) It should be noted that this approach is related to monitoring the modal participation factors for the particular problem.

(2) Regardless of the number of modes chosen, it is probably a good idea to rerun the reduced analysis with more modes, in order to compare the original simulation to this "refined solution". If no major changes are found in the results, there is some hope that the reduced solution exhibits the most important mechanical behavior of the model problem. When the reduced method is used for the purposes of an inexpensive preliminary design, an unreduced solution can also be used for comparison purposes.

(3) If the problem being solved is linear, then the reduced stiffness can be diagonalized once, and then the cost of performing each step of the algorithm is negligible, since the matrices need not be reformed at each step, and no equations need to be solved. In this case, it is inexpensive to add modes, and so refined analyses with more vectors in the projection basis are a practical approach to checking for convergence.

Finally, the question of what can go wrong with the scheme should be addressed. As will be seen in the next chapter, if the physical interpretation of the projection space is inconsistent with the physical interpretation of the unknowns sought in the temporal integration scheme, poor results can be obtained even when the (projected) residual norm is small. In addition, if many modes are considered in a large problem, the formation of the reduced matrices may involve appreciable computational error. For this reason, it is recommended that the evaluation and solution of the reduced problem be carried out in

higher-precision arithmetic, if possible. For instance, if the element stiffness matrices are evaluated as 32-bit reals, then it is appropriate to store and manipulate the reduced matrices Q^TAQ, Q^TMQ, and Q^Tf as 64-bit floating-point numbers. Element assembly for an unreduced formulation merely involves summing the element terms into the global arrays, but for a reduced formulation, it involves the formation of many matrix inner product multiplications. It is well known that accumulation of inner products in double precision is an inexpensive means for reducing the propagation of round-off error, and it is clear that as long as the reduced matrices are relatively small, the cost of solving these equations in double-precision arithmetic is not a serious computational burden.

Comparison with Related Algorithms

The use of modal methods in linear dynamics has a rich history, including the standard analysis technique known as "normal mode analysis". The use of Lanczos vectors and the application of modal methods to nonlinear problems is a more recent development, but there are still dozens of recent references in this field. This section will therefore be confined to the consideration of only several related publications, divided into three general topics:

(1) "classical" mode-superposition for nonlinear problems
(2) Lanczos and related schemes for linear problems
(3) Lanczos and related schemes for nonlinear problems

What might be termed "classical" mode-superposition techniques involve the use of eigenvectors of the linearized problem for generating approximate solutions for the nonlinear model. This approach is taken by several authors, including (Nickell, 1975), (Bathe and Gracewski, 1981), (Geschwinder, 1981), and others. A more recent work (Idelsohn and Cardona, 1985) considers the use of eigenvectors combined with "modal derivatives", and includes some emphasis on updating the modes during the calculations.

The use of Lanczos vectors in an inverse (non-shifted) setting has been explored for linear problems by (Wilson, et.al., 1982) and (Nour-Omid and Clough, 1984) for general linear problems. Wilson's reference includes an ingenious rediscovery of the Lanczos Algorithm from a physical standpoint of what might be termed "neglected inertial forces". Trying to account for these neglected forces leads naturally to the Krylov Sequence and to an

algorithm that is roughly equivalent to the Lanczos Algorithm with complete reorthogonalization. Nour-Omid and Clough present similar results in the framework of the Lanczos Algorithm, and use the more efficient three-term recurrence to generate a basis for the projection space. Several papers involving this approach to different problems (Bayo and Wilson, 1984a), (Bayo and Wilson, 1984b) and (Wilson and Bayo, 1986) have also appeared in the literature. Since all these problems are linear, the reduced equations of motion can be diagonalized once, and then many of the computational considerations of the last section do not apply. In particular, the problem of consistency between the projection basis and the Newmark unknowns does not appear to be a concern when the equations are expressed in a diagonal form.

The use of Lanczos and related methods on nonlinear problems is a more difficult topic. Application of the Krylov Sequence to solve a linear problem (one resulting from one step of a nonlinear iteration scheme, for instance) can be shown (Golub, 1985) to lead to the well-known Conjugate Gradient Method for the iterative solution of positive-definite linear systems. Alternatively, these calculations can be rearranged so that the projection solution involving the Krylov Sequence yields the Lanczos Algorithm in a direct (i.e. non-inverse) setting. This approach is taken by (Nour-Omid, Parlett, and Taylor, 1983) to develop a Newton-Lanczos procedure for the solution of nonlinear problems. Newton's Method is used to linearize the nonlinear model, and the resulting system of linear equations is solved via a Lanczos Method. This approach is obviously related to the proposed algorithm, differing in that the projection basis is constantly updated, and in the fact that the Lanczos Algorithm is used in a non-inverse setting. Applying the Lanczos algorithm in this fashion allows the projection space to be recalculated at each step (so that the projection space is always associated with the current pencil (K(t), M(t)), instead of the pencil associated with some earlier time step), but leads to two disadvantages. First, the projection basis is associated with the largest eigenvalues, which are the least important in the approximation of the solution (recall the Fredholm Alternative Theorem). In simple terms, this approach converges from the "wrong" end of the spectrum, which might be expected to cause difficulties if the coefficient matrix is ill-conditioned. Second, the Krylov Subspace is invariant with respect to a shift, so this approach cannot simply incorporate a frequency shift for the identification of resonance modes in the approximate solution. Nonetheless, the algorithm presented appears to be efficient and reliable, and performs well on a variety of problems, including a interesting nonlinear problem with a singular stiffness matrix. In addition, this approach can be applied to indefinite systems, so it appears to show much promise for the practical solution of many important nonlinear problems.

Chapter 4

Example Problems and Results

Chapter 4: Example Problems and Results

Introduction

The reduced coordinate algorithm developed in the last chapter was motivated by considerations of accuracy, computational efficiency, and minimization of storage. In theory, it appears that the proposed method would be useful for solving many large problems, but the true test of any algorithm is found in how well it actually solves practical engineering problems. The verification of any algorithm must therefore eventually include the solution of "real-world" problems.

The purpose of this chapter is to consider the solution of two representative problems in dynamics, using both an "unreduced" formulation (direct step-by-step integration of the full equations of motion) and the proposed reduced coordinate algorithm. These two problems are relatively simple ones, so that the behavior of the solution can be evaluated easily and the algorithms compared without unnecessary confusion. Although the problems are not as complex as many in mechanics, they are not contrived. In fact, this simplicity will be seen to be a serious handicap for the reduced method, so that good performance on these simple problems can be taken as evidence of even better results on larger, more intractable models.

The problems considered in this chapter involve free vibration of elastic solids subjected to initial disturbances. The first problem is that of a linear beam whose free vibrations are constrained by a nonlinear support. This beam problem is essentially one-dimensional, so that the solution history for the entire beam can be displayed in one three-dimensional plot involving space, time, and displacement. This characteristic makes it easy to compare the methods used. The second problem involves the free vibration of a relatively stiff building that is founded in a softer soil deposit, and subjected to a blast loading. This model is much more typical of most problems in engineering mechanics, as it involves larger systems of equations, larger bandwidths, and dissimilar materials. The effect of the type of integration scheme used is the most important aspect of this problem, and since the geometry is two-dimensional, the solution history is much harder to visualize.

Extensions of the reduced coordinate algorithm to more difficult problems including material nonlinearities and dissipative (as opposed to dynamic) time-dependence can be found in (Mish, 1987). Extensions of this method to include multiple phases (soil-fluid-

structure problems) and alternative projection bases are discussed in Chapter 5 of this document.

Beam with Nonlinear Support

The first problem considered involves the free vibration of the beam shown in Figure 4.1. This uniform beam is initially displaced into the shape given by the first mode of vibration, and released at time t = 0. After one quarter of the beam's fundamental period has elapsed, the free end of the beam comes in contact with the support, and the displacement field becomes a much more complicated function of time. Four different solutions are considered:

(1) Direct integration of the unreduced problem with a discretization involving 10 beam elements and a time step of 0.01 second.
(2) Direct integration of the unreduced problem with 20 elements and a time step of 0.005 seconds.
(3) Reduced algorithm using three Lanczos vectors and the same data of solution (1)
(4) Reduced algorithm using three Lanczos eigenvectors and the data of (1)

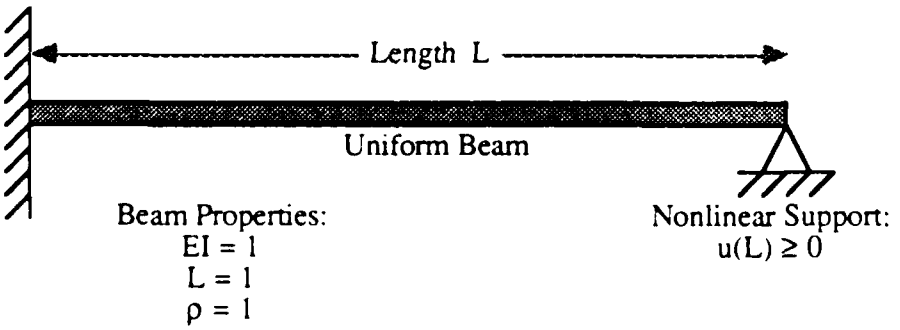


Figure 4.1: Beam Geometry and Properties

The transverse displacement of the tip of the beam is shown in Figure 4.2. Note that all the methods used show exact agreement until the barrier is hit by the tip of the beam, and

relatively similar results afterwards. In this problem, a Newmark scheme involving incremental accelerations was used in conjunction with a projection basis with the physical interpretation of beam displacements. This choice was made to realize the simple iteration scheme of the acceleration form of Newmark's method, while maintaining the interpretation of the Lanczos vectors as generalized displacement coordinates. In general, this choice might be expected to lead to some numerical difficulties (as will be seen in the next example in this chapter), since it amounts to forcing the incremental accelerations at each step to have the shape of the estimates for the lowest eigenvectors of the displacement-based problem. However, this beam problem has so much required continuity (continuity of displacement and its first derivative) that this inconsistency is not a problem, and the reduced solution is well-behaved. This can be taken as a demonstration of the fact that some "well-behaved" problems do not require the sort of considerations of consistency (between primary unknowns in the temporal integration and primary unknowns in the generation of projection vectors) that were discussed in the last chapter.

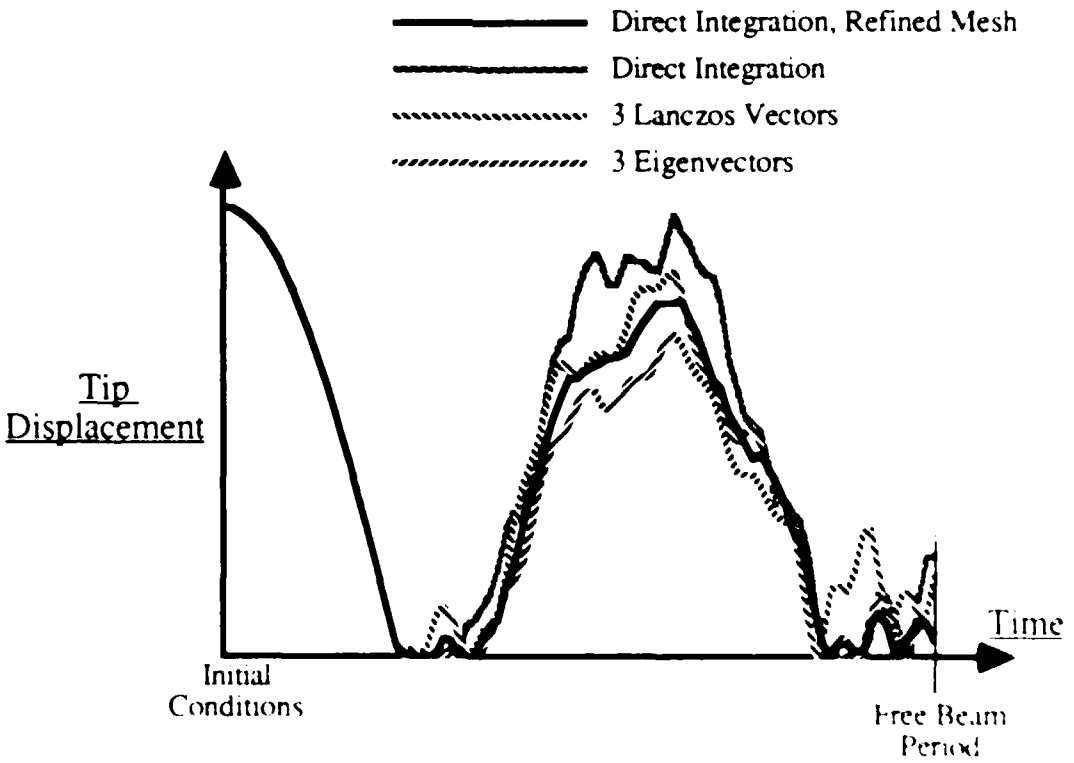


Figure 4.2: Tip Displacement History

Figures 4.3 through 4.6 show the behavior of the four solution histories. Note that the reduced solutions show a little less high-frequency behavior (since the higher frequencies have been filtered by the projection scheme), and thus more closely approximate the "smoother" solution obtained by subdividing the spatial and temporal discretizations.

In general, the reduced coordinate algorithm required about twice as much computational work as the unreduced formulation for this problem. While that may seem like poor performance, it is actually very efficient, given the analysis of the last chapter. This problem is one-dimensional, and so the bandwidth is very small (half-bandwidth = 4) and does not grow as the mesh is refined. Thus, in a one-dimensional problem like this one, the cost of solving the unreduced system of equations grows linearly with the number of nodes. The reduced algorithm cannot compete in this setting, since it is designed to be efficient for problems where the bandwidths are large and grow larger as the mesh is refined. (The next problem is a better example of this sort of growth.)

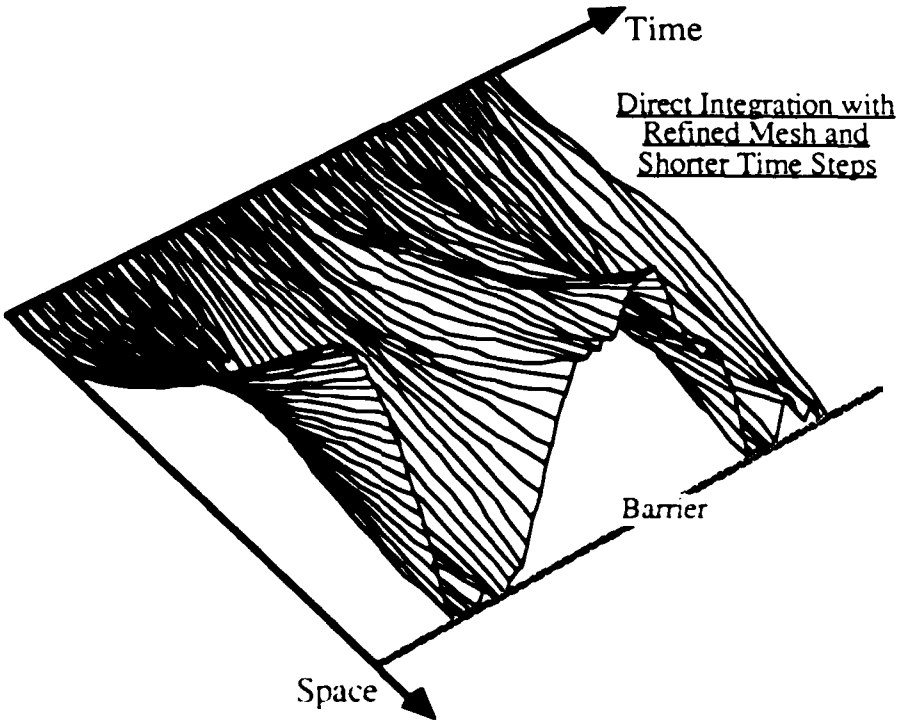


Figure 4.3: Results of Refined Unreduced Solution (20 Elements)

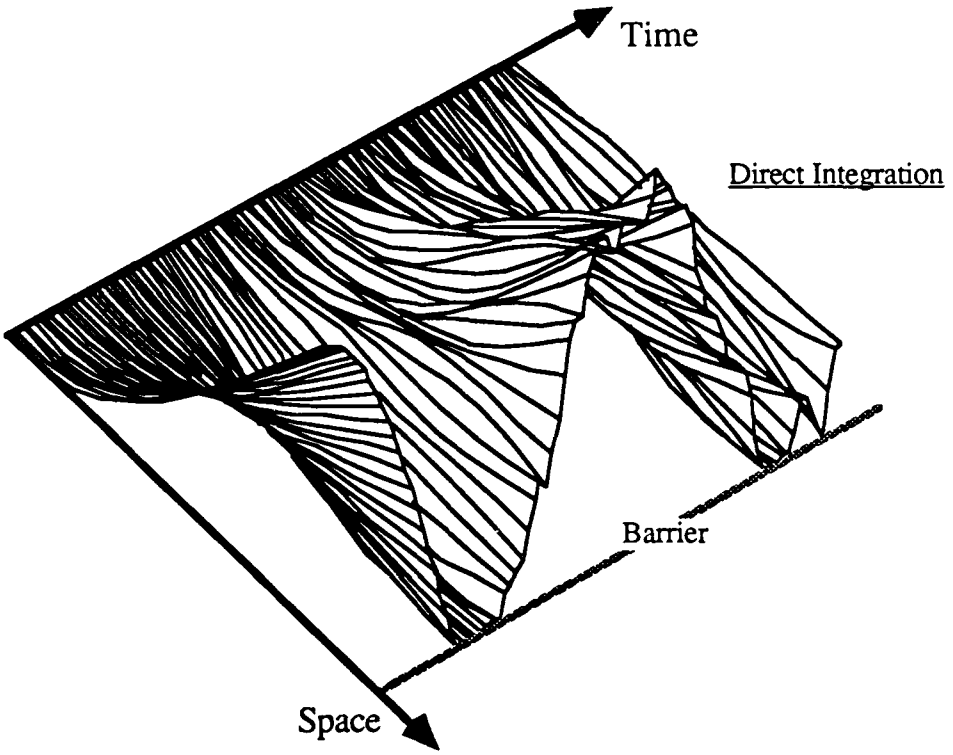


Figure 4.4: Results of Unreduced Solution (10 Elements)

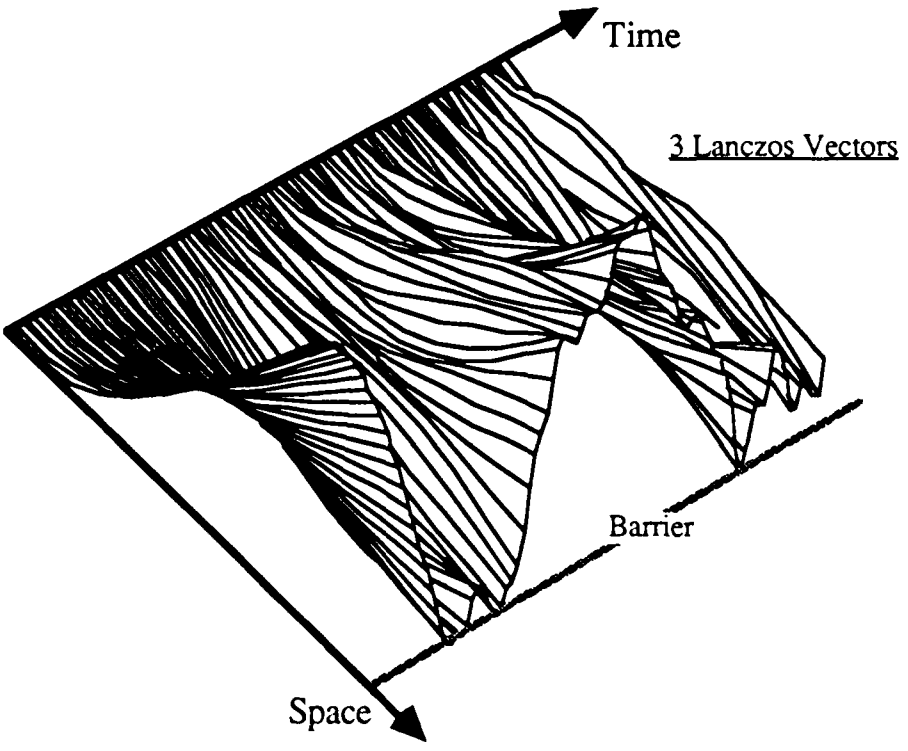


Figure 4.5: Results for Lanczos Vector Reduction (10 Elements)

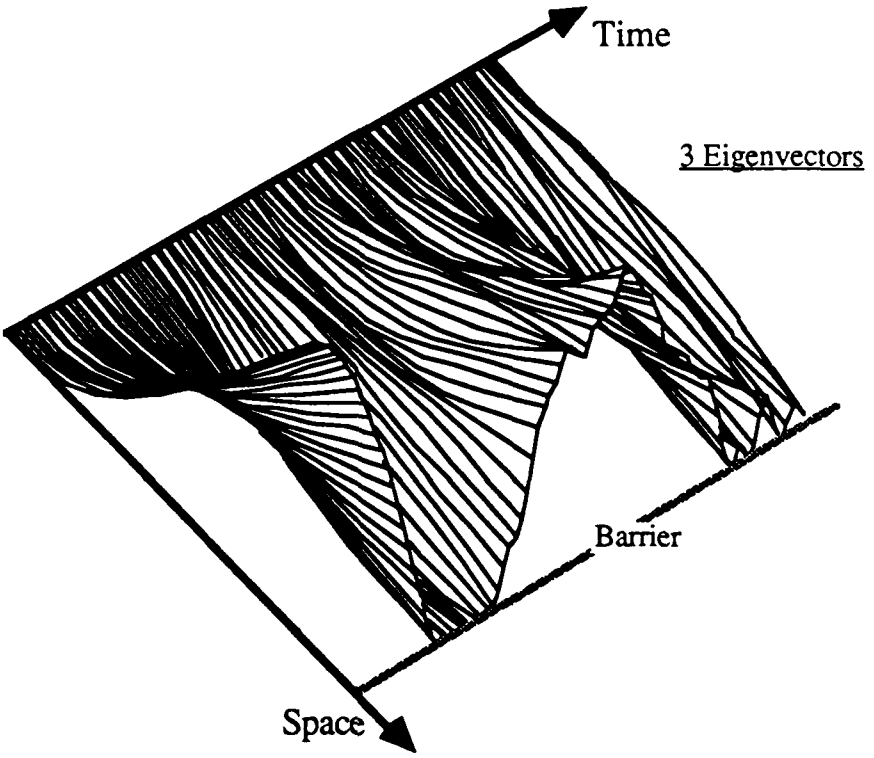


Figure 4.6: Results for Lanczos Eigenvector Reduction (10 Elements)

Blast Load on Soil-Structure System

The second problem considered involves the dynamic response of a building founded in relatively soft soil. The building is loaded by a blast pressure of 2 psi for a duration of ten seconds. In the first five seconds, the blast load is uniform, and in the second five the load decreases linearly from 2 psi back to zero. This problem is the first step in a more complex analysis in (Mish, 1987) involving the use of bounding surface plasticity for the modelling of the nonlinear behavior of the soil. For the problem considered herein, both the soil and the building are taken to be linear elastic, with material properties given by:

E_{soil} = 300 ksi	ν_{soil} = 0.2	ρ_{soil} = 0.125 ksf
E_{bldg} = 7500 ksi	ν_{bldg} = 0.18	ρ_{bldg} = 0.150 ksf

The geometry of the problem is shown in Figure 4.7, and the appropriate boundary conditions are illustrated in Figure 4.8 (this figure also shows the location of two particular elements that are used to monitor the behavior of all the analyses).

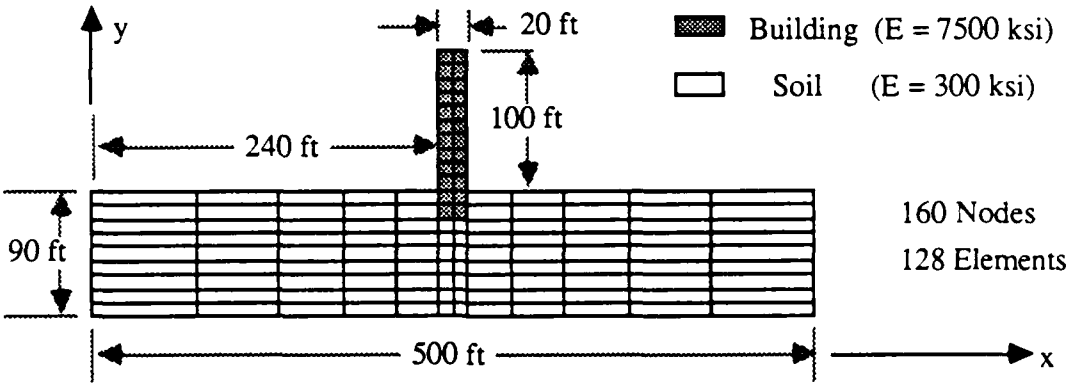


Figure 4.7: Blast Problem Geometry and Materials

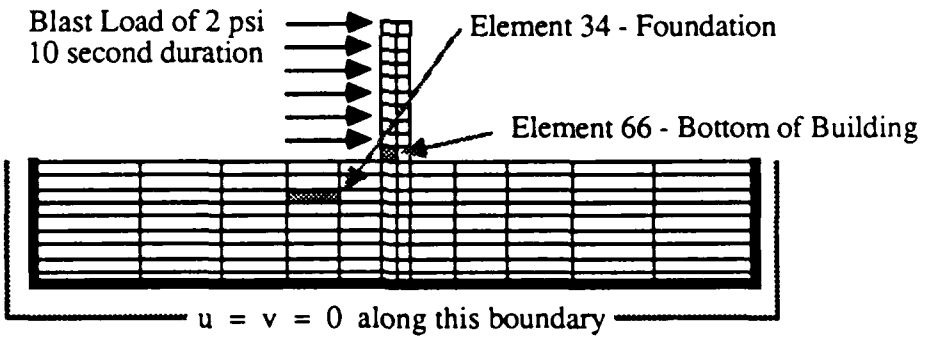


Figure 4.8: Boundary Conditions for Blast Problem

Three types of analysis are considered for this particular problem:

(1) The direct solution of the full equations of motion using an unreduced formulation. In this case, incremental accelerations are the primary unknown in the Newmark iteration.

(2) A reduced formulation using Lanczos vectors from the displacement form of the problem for a projection space, applying the incremental acceleration form of Newmark's Method (as in the beam problem considered earlier).

(3) A reduced formulation using the same Lanczos vectors as in analysis (2), except using a Newmark-Newton iteration involving incremental displacements as the primary unknowns.

In cases (2) and (3), anywhere from one to six Lanczos vectors were used, and all of these vectors were generated using the initial acceleration as a starting vector. This choice corresponds to choosing the initial mode to correspond to the pseudo-static response to the intial applied force. Results will be reported in the following interpretation for from one to three vectors. Adding vectors beyond a three-dimensional subspace did not appreciably increase the accuracy of the analysis, but began to increase the computational effort. In this problem, the reduced algorithm typically used about half as much computer time as did the unreduced problem (which is not surprising since the size of the equation set and the associated bandwidth are much larger than in the beam example). In addition, most of the computer effort for the reduced solution involved the computational overhead of formation of stiffness matrices, which were evaluated at each step for this linear problem in order to model more accurately the performance of these schemes in a nonlinear setting. The existence of this computational overhead manifested itself in the fact that the increase in size of the reduced problem from one to three vectors produced only a marginal change in the computational effort required for the analysis. Given the operations counts considered in the last chapter, it appears for two- and three-dimensional problems, the reduced algorithm will be very inexpensive compared to unreduced solution techniques. It should be noted that this problem is only a coarse mesh for a relatively small two-dimensional problem. It is expected that this reduced algorithm will soon be used by the authors on problems that are orders of magnitude larger than this one. Finally, another expense for the reduced method is that the code used to do the reduced problem is heavily instrumented for purposes of evaluation of the algorithm. (For instance, the reduced program does most of the work of the unreduced one in order to keep "two set of books" for comparisons of the methods while the proposed algorithm is being modified and optimized.) The program used to solve the unreduced problem is much closer to a "production code" than the one used for the reduced algorithm.

The shapes of the displacement fields for the first three Lanczos vectors ("modes") are shown in Figure 4.9. These patterns of displacement are magnified a few thousand times

because the normalization of these modes with respect to the mass matrix yields actual displacement components that are on the order of thousandths of a foot. As may be noticed in Figure 4.9, the shape of the displaced building in the first mode appears to corresponds to a rigid rotation of the building under the applied load, and the next two modes demonstrate some bending behavior of the building.

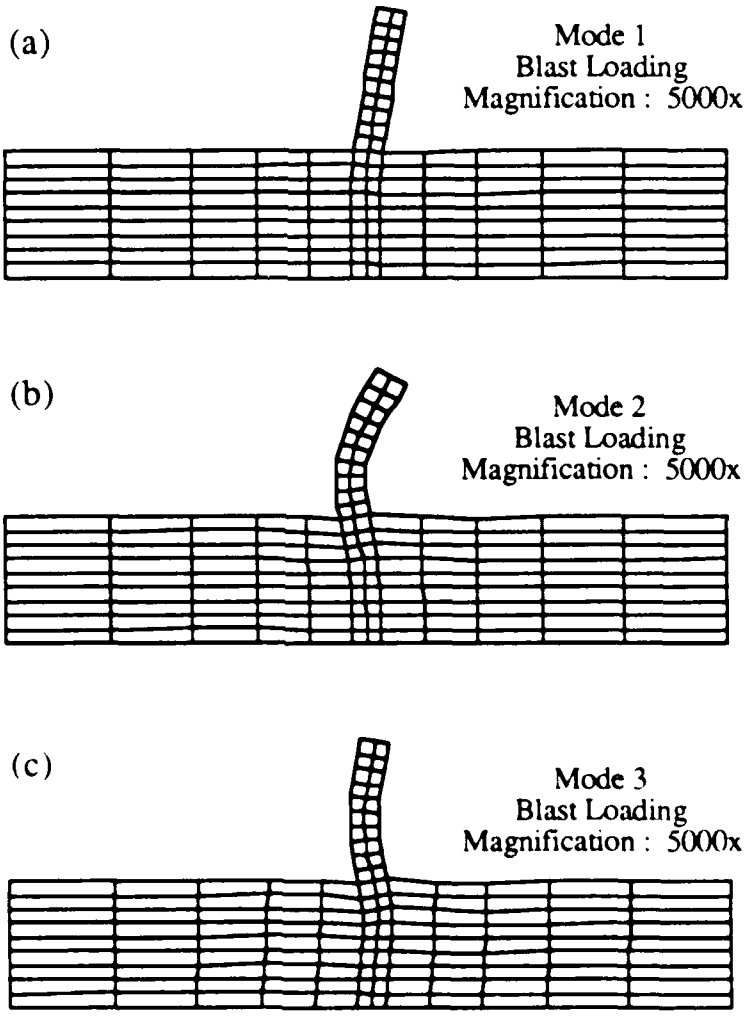


Figure 4.9: Modes for the Blast Problem

In order to more clearly see the stress state corresponding to these modes, the normal and shear stresses for the first three modes are shown in Figures 4.10 - 4.12. Figure 4.11 clearly shows the distribution of bending stress in the building, with the first mode

corresponding to the building acting as a cantilever, and the second and third modes showing bending stresses more characteristic of a simply-supported beam. The shear stresses shown in Figure 4.12 also demonstrate this result.

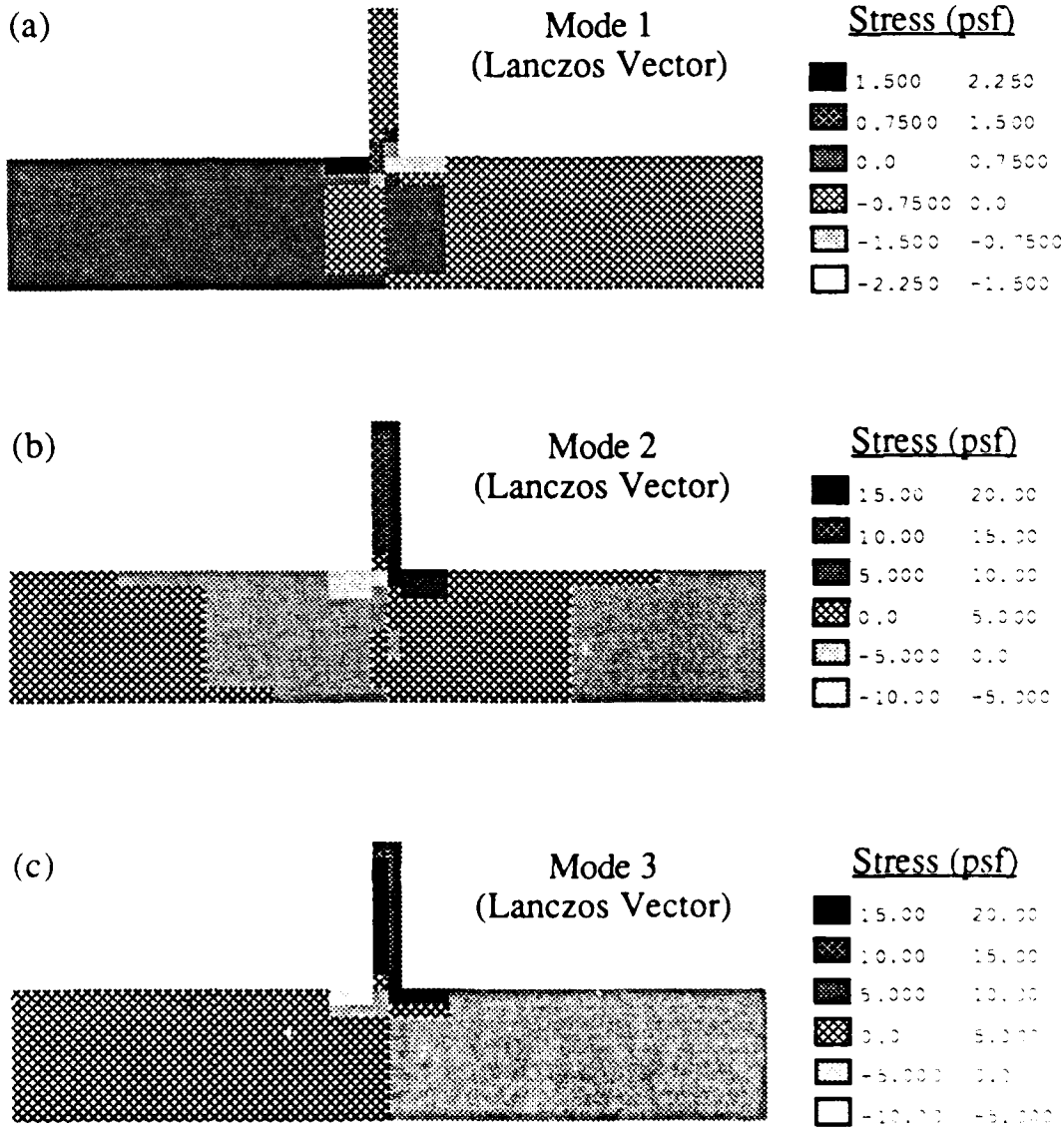


Figure 4.10: Modal Stresses σ_x for the Blast Problem

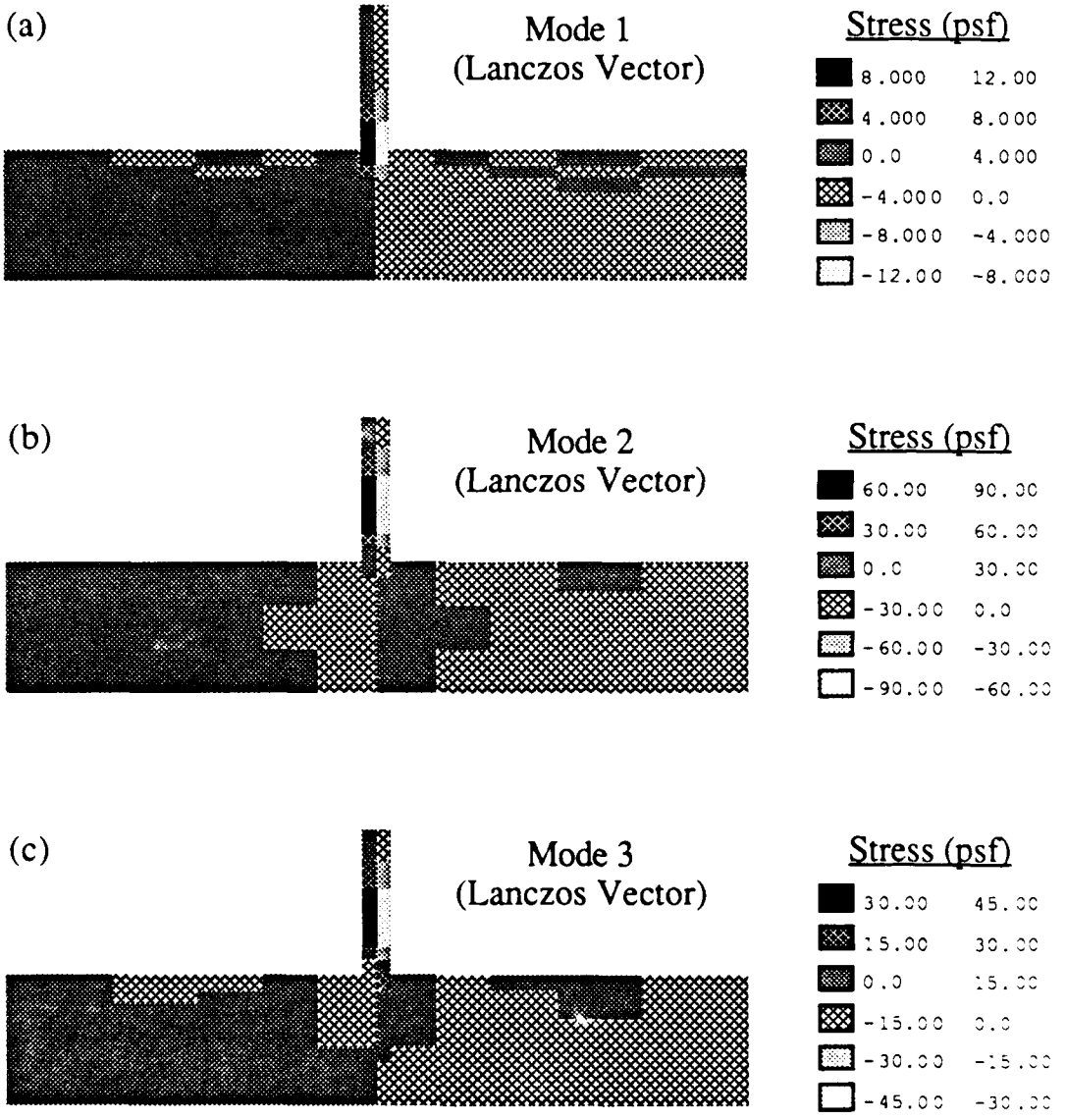


Figure 4.11: Modal Stresses σ_y for the Blast Problem

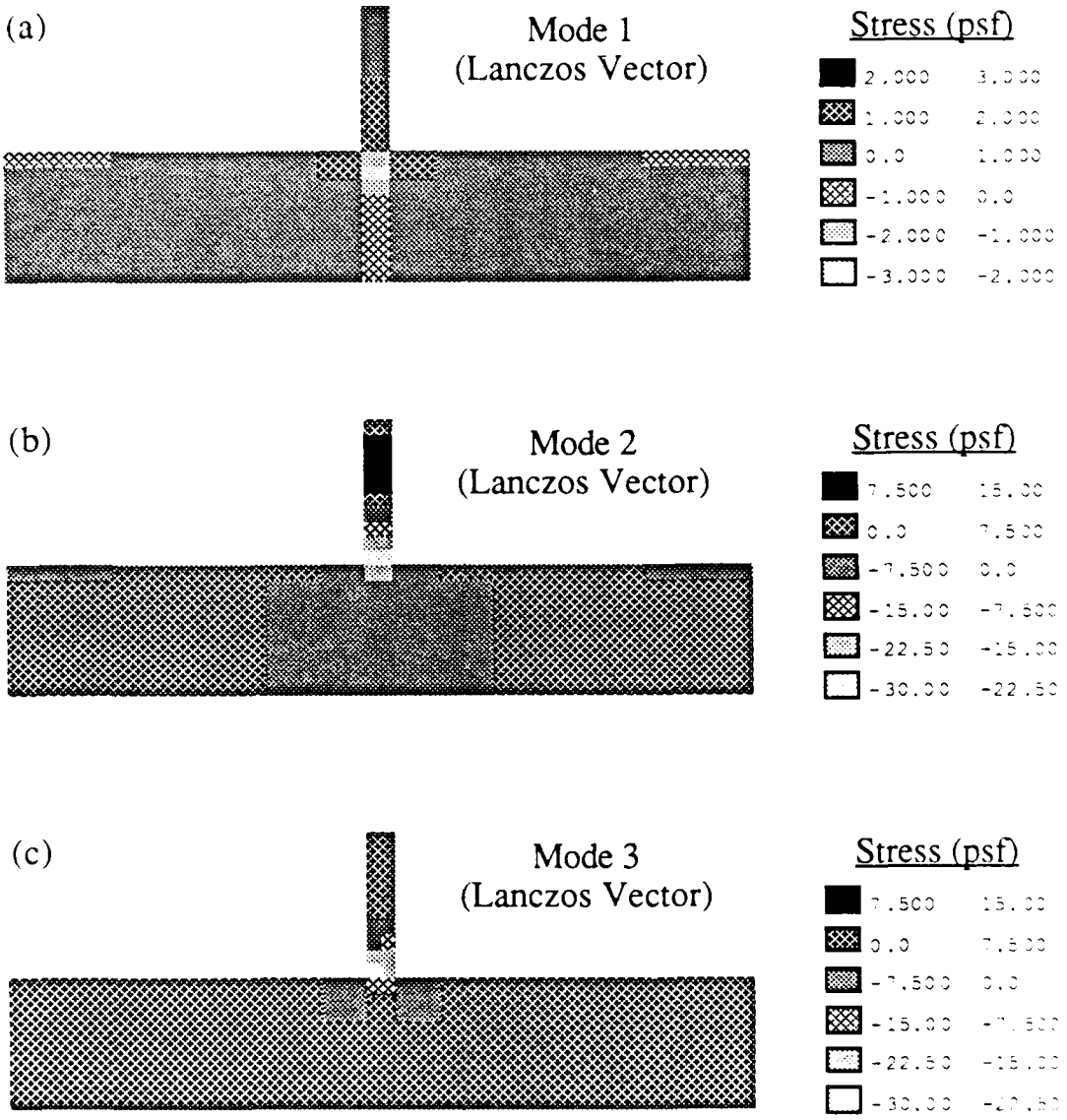


Figure 4.12: Modal Stresses τ_{xy} for the Blast Problem

The horizontal displacements at the mid-height and the bottom of the building are shown in Figure 4.13. The solution history illustrated is for a forty-second period showing the ten-second blast and the following thirty seconds of free-vibration. The solutions for the unreduced problem and for all the reduced problems involving incremental displacements are essentially coincident, so these reduced results are not plotted. The solution for the reduced problem using one mode and incremental accelerations as primary unknowns

shows a pronounced "drift" with time. This drift gets worse as modes are added, so the results for a larger set of Lanczos vectors are not shown. What is happening here is that the incremental accelerations at a given step are not particularly similar to the displacement projection basis, and so the projection solution obtained (while satisfying the equilibrium equations in a projected sense) is not very accurate in the setting of the larger problem. In the beam problem, the more stringent continuity requirements of the model blurred the distinction between the approximate subspace for the solution and the type of unknown solved for in the temporal integration scheme. In this problem, the obvious interpretation is that the temporal integration scheme must use a solution that is consistent with the approximate solution space.

Results involving stress histories in the elements highlighted in Figure 4.8 are shown in Figures 14-18 (element 34, in the soil layer), and in Figures 19-23 (element 66, at the bottom of the building). Note that Figure 15 (the history for the reduced method using a standard Newmark iteration) shows the same pronounced drift in the oscillatory stress states. Figure 19 shows that, for the bottom of the building, the entire solution has been drowned out by this drift. In this case, forcing the acceleration to take the shape of the displacement modes guarantees huge displacements and stresses for the solution history. This clearly shows that care must be taken to use modal projection methods with appropriate time-stepping schemes, especially in nonlinear settings where the reduced problem cannot be diagonalized.

In conclusion, it should be reiterated that the reduced coordinate scheme has produced accurate answers on these example problems, and that it can be implemented relatively easily in many Finite Element codes. The algorithm appears to show a great deal of promise for the solution of large problems, especially since it is competitive with standard methods on smaller problems (such as these two) where an unreduced formulation has a computational advantage.

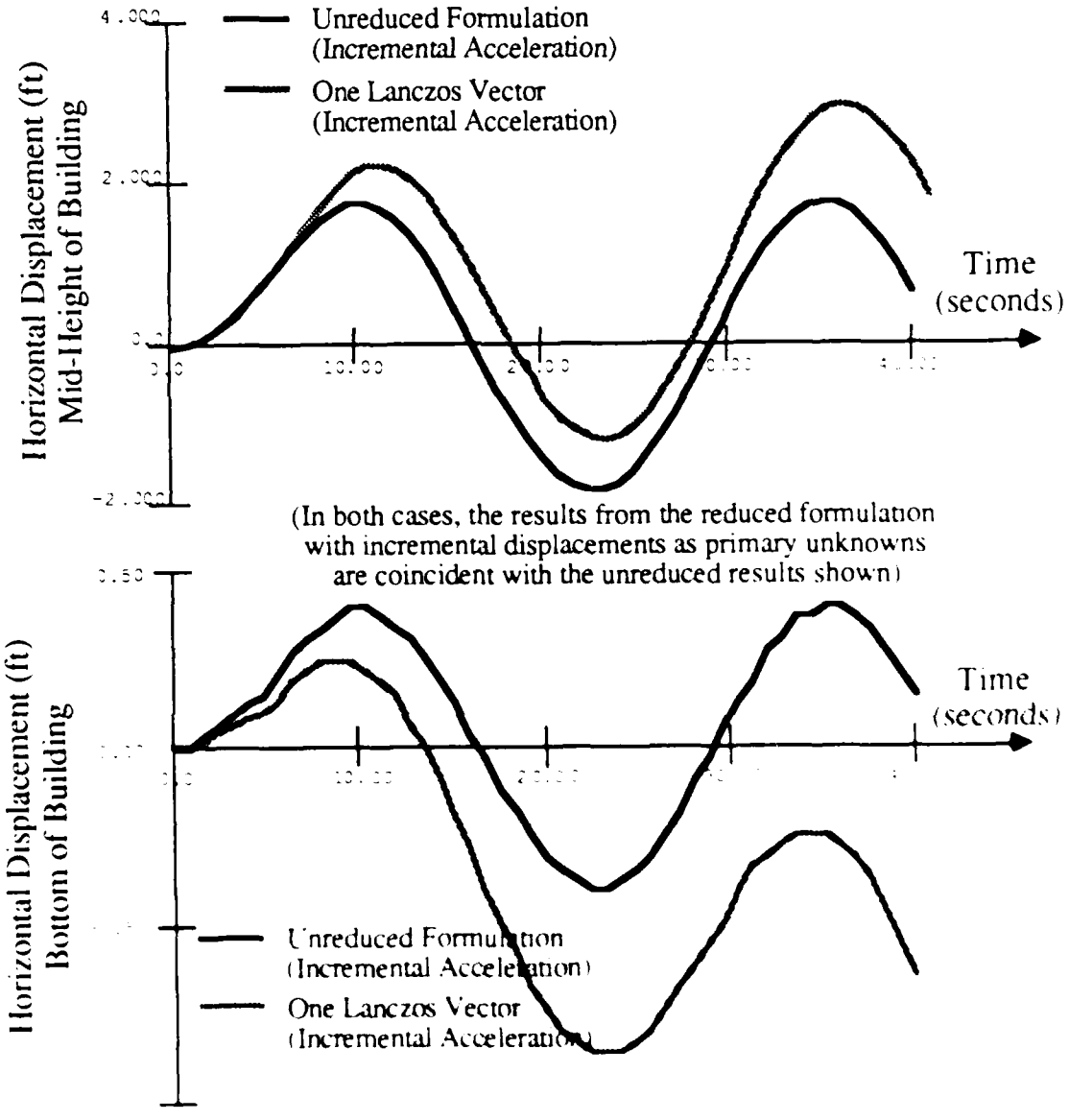


Figure 4.13: Representative Horizontal Displacements

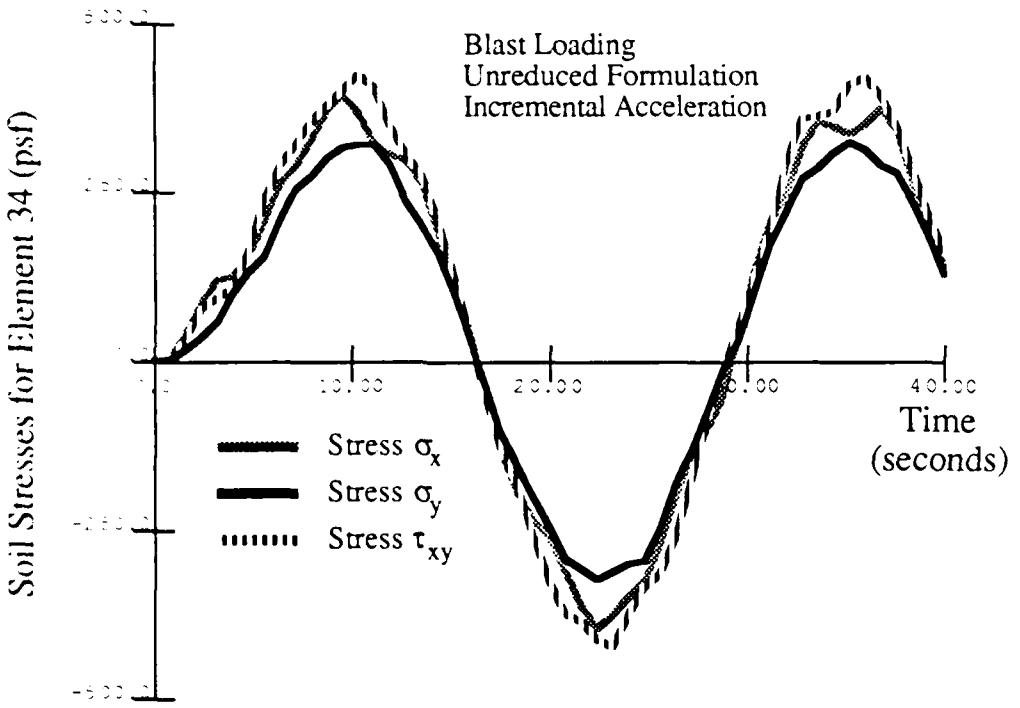


Figure 4.14: Element Stresses for Unreduced Problem

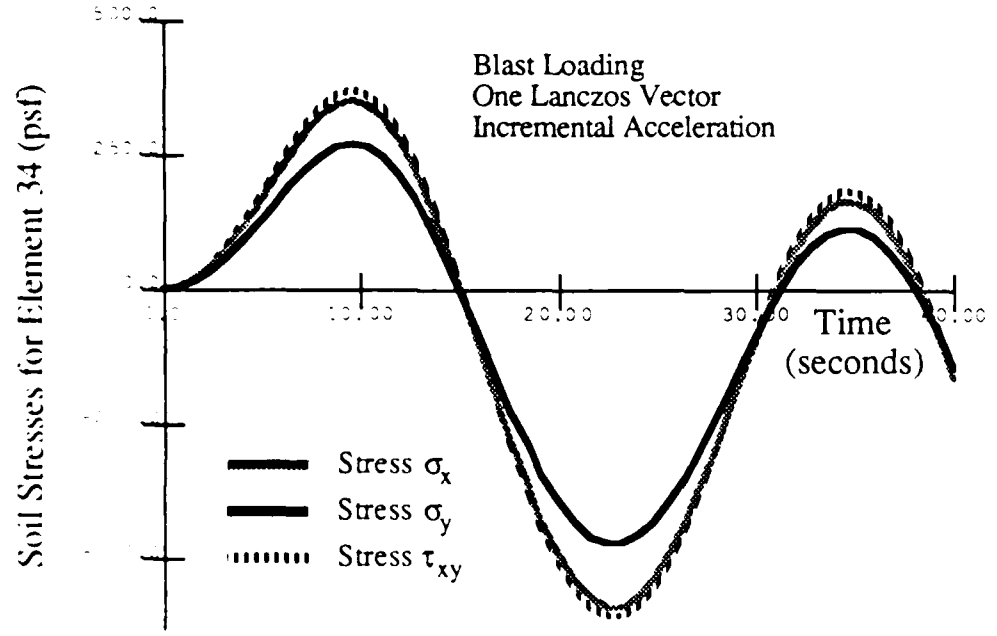


Figure 4.15: Element Stresses for Inconsistent Reduction

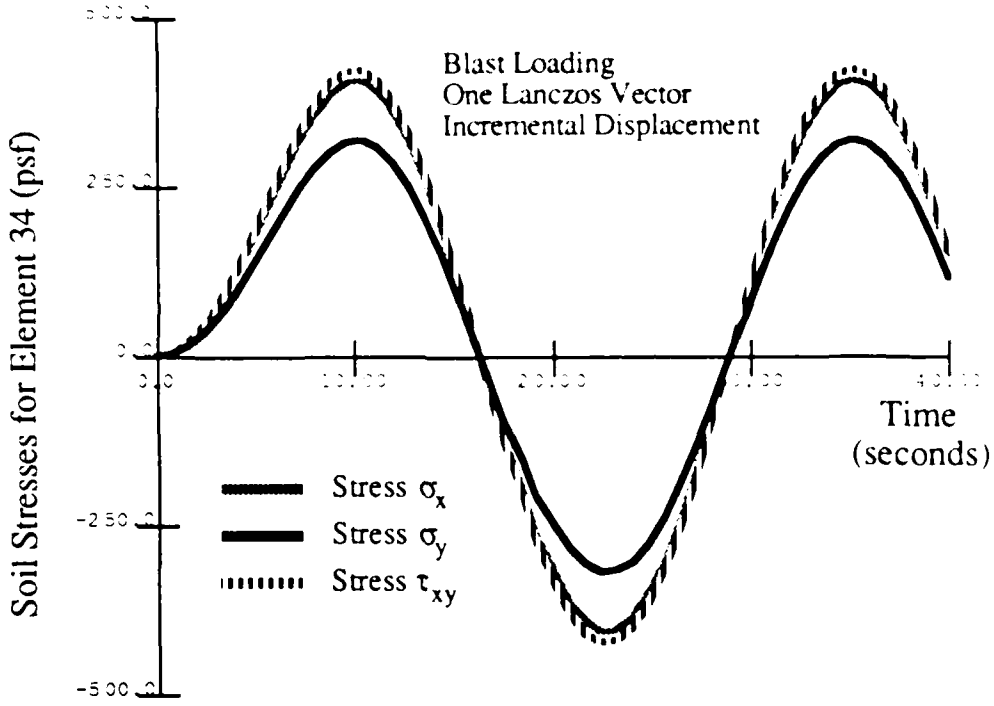


Figure 4.16: Element Stresses for Reduced Problem (1 Mode)

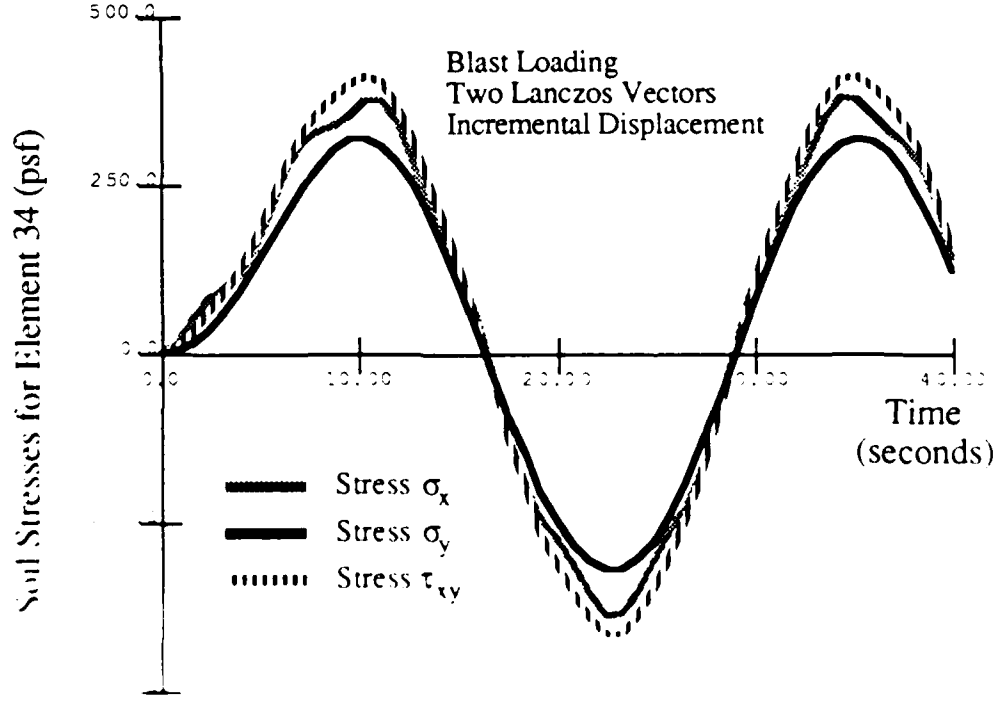


Figure 4.17: Element Stresses for Reduced Problem (2 Modes)

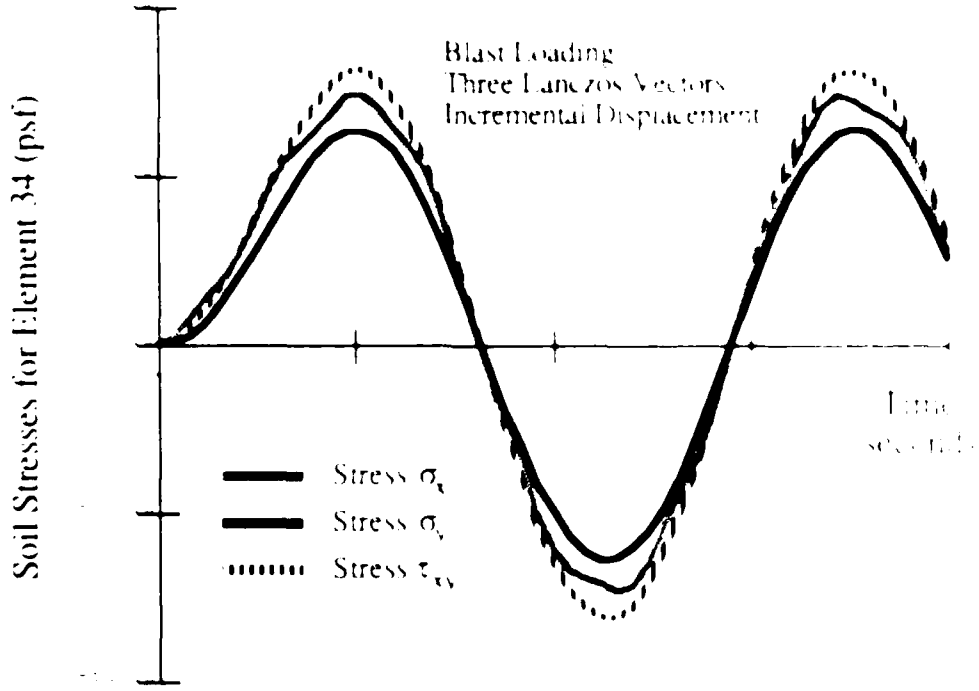


Figure 4.18: Element Stresses for Reduced Problem

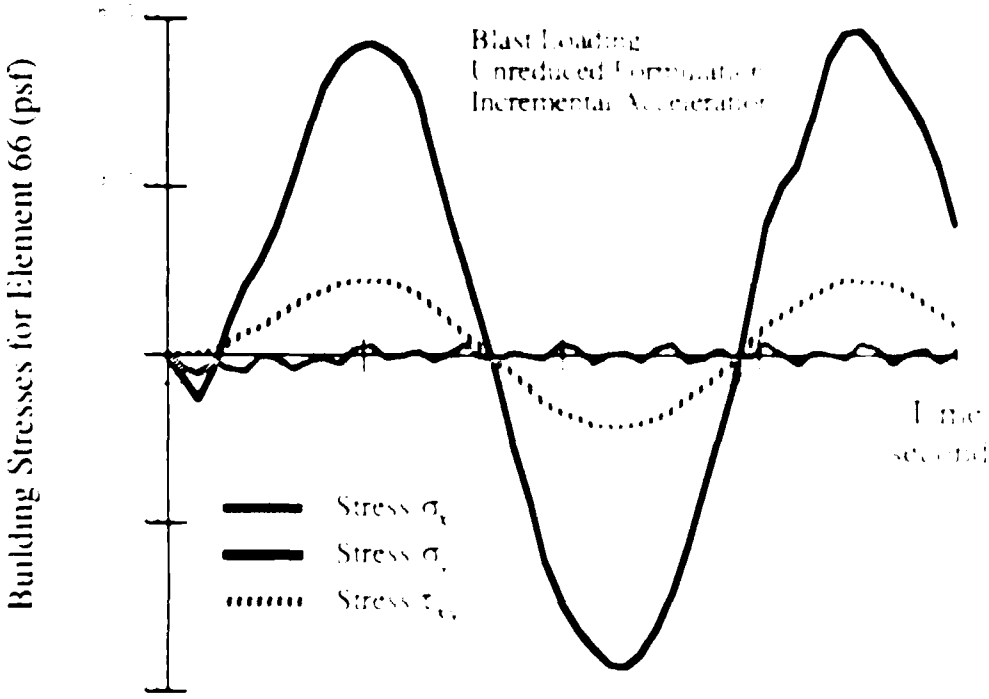


Figure 4.19: Element Stresses for Unreduced Problem

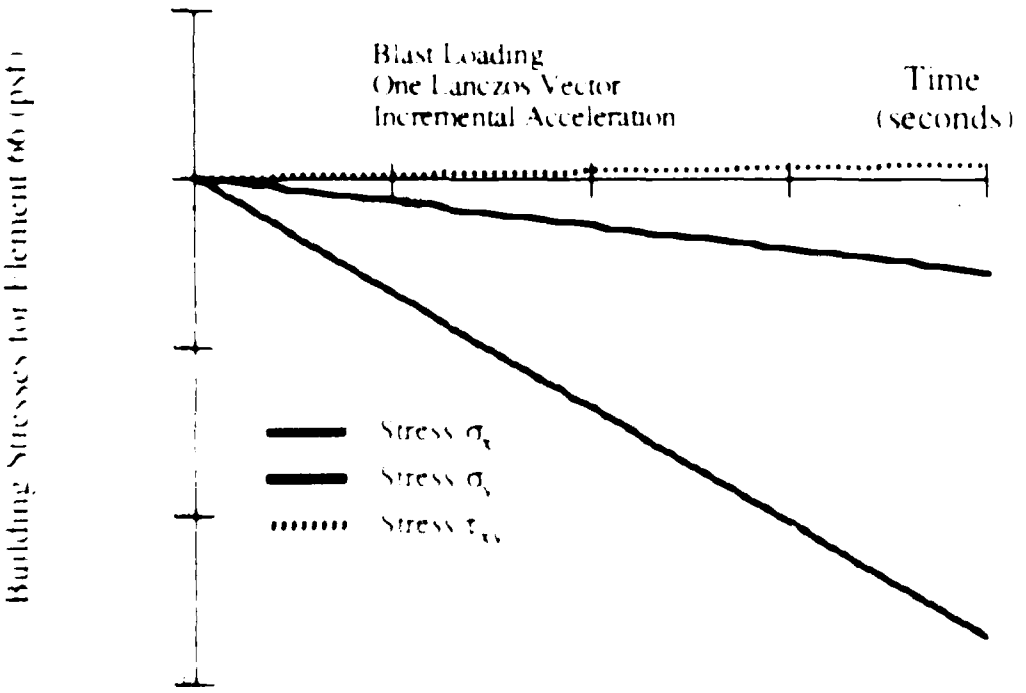


Figure 4-20. Element Stresses for Inconsistent Reduction

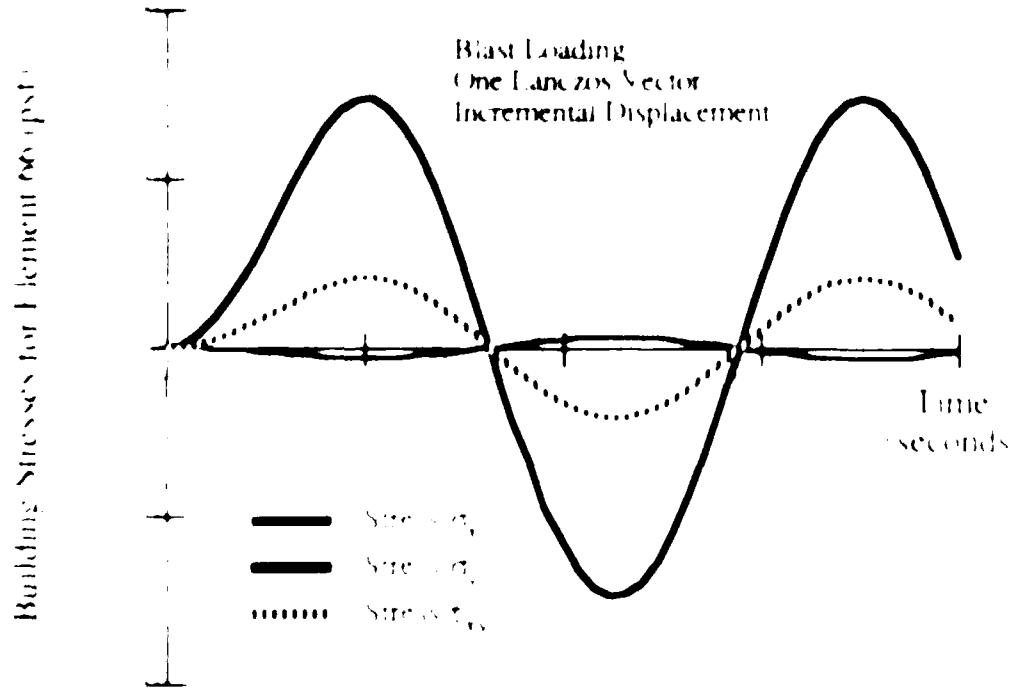


Figure 4-21. Element Stresses for Reduced Problem 3 Mode

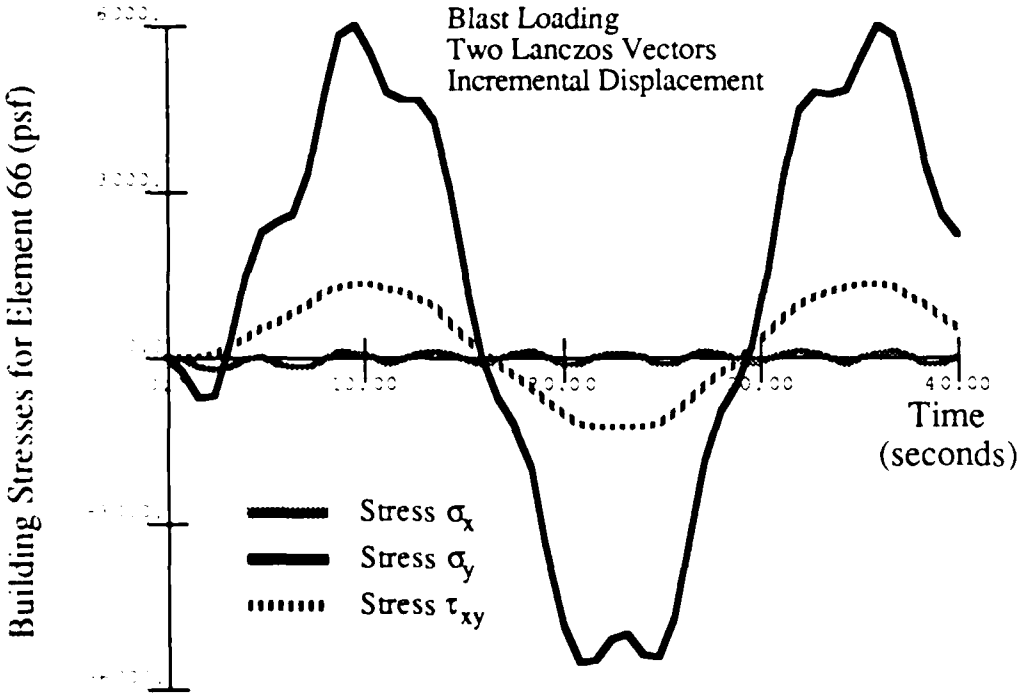


Figure 4.22: Element Stresses for Reduced Problem (2 Modes)

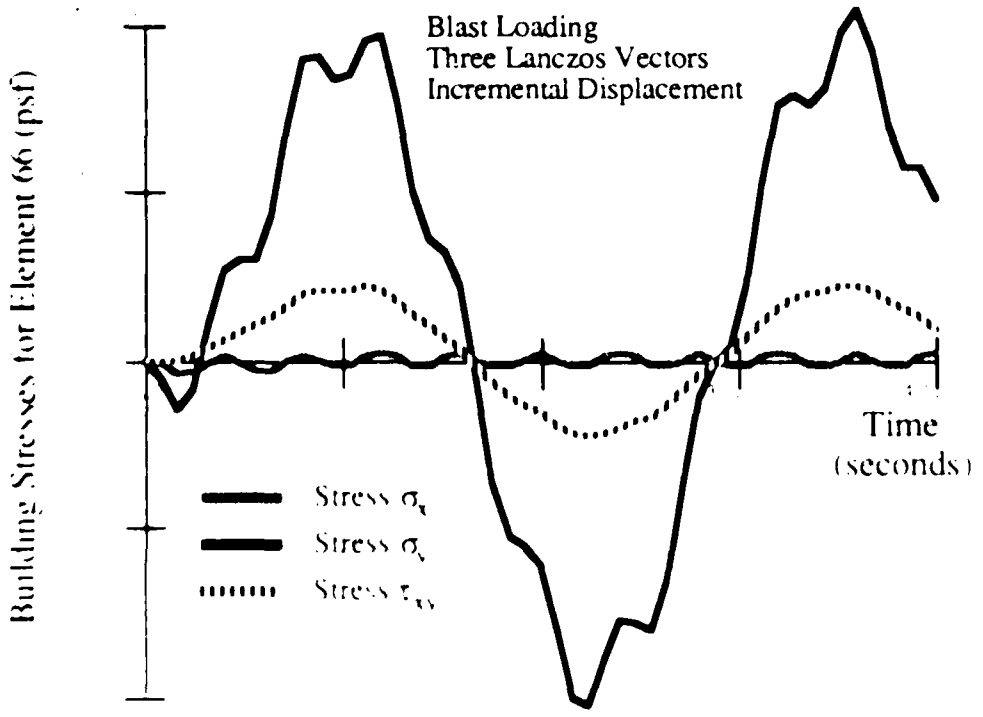


Figure 4.23: Element Stresses for Reduced Problem (3 Modes)

Chapter 5

Conclusions

Conclusions

The proposed research has been concerned with the reduction of large mechanical problems to a more manageable size. The results of the last chapter show that the reduced coordinate algorithm suggested in this document can be used to achieve this reduction, while preserving the important mechanical behavior of the original model. Through the use of this method, many large models can be solved on smaller machines, and presently intractable problems can be solved approximately on larger computers. The proposed reduced coordinate algorithm represents an attractive alternative formulation to direct methods for solving many important problems in mechanics.

Suggestions for Further Research

The derivations and results presented in this document represent an attempt to develop and apply this proposed algorithm to some large problems in mechanics, but these results are only a first step towards a more general method that can be used on a wide variety of problems. In this section, some suggestions for future work will be outlined and discussed.

(1) Reduced coordinate analysis of problems involving soil-fluid-structure interactions.

The original goal of this ongoing research was to model large problems that were intractable using a direct (i.e. non-reduced) approach. An excellent example of this type of problem involves the vibration of a structure that rests upon a saturated soil. The dynamic behavior of this system involves the solid mechanics problem of a soil-structure system, but this behavior is tightly coupled to the flow problem of water in the pores of the soil. If the soil permeability is small, then some simplifying assumptions about the relative displacement of the soil and fluid may be made, but the low permeability and high bulk modulus for the soil-water system render the system nearly incompressible, which results in a difficult numerical problem. Conversely, if the soil is very permeable, the incompressibility is less of a concern, but the relative displacement of the solid and fluid complicate the kinematics of the problem. In either case, an accurate analysis can be computationally very expensive.

The modal algorithm can be used to reduce the size of this type of problem to a more

manageable level. It is proposed to use the reduced algorithm on several large soil fluid structure problems to determine whether it performs well in this setting, and to find out what type of reduced coordinates work best. This research will require considerable basic work to be able to model the physics of the problem, as well as recasting the resulting direct analysis into an appropriate reduced form. Previous research into the nonlinear material behavior of soils with low permeability, as well as similar ongoing research for granular soils, will be used to model the response of the soil, and this response will be the primary source of nonlinearities in these problems.

(2) Generalization to different types of modes.

The research to date has been developed theoretically from the standpoint that an orthogonal set of modes is available for projection of the large problem onto a smaller solution subspace. In the research completed to date, these modes have included exact eigenvectors and Lanczos vectors for some particular state of the matrix equation of motion. It is proposed to consider what other modes might be used for a solution, and whether different parts of a coupled problem might require a "mixing" of types of modes. It may also be helpful to include modes that represent generalized derivatives of the eigenvectors or Lanczos vectors. Some research has been done along these lines (Idelsohn and Cardona, 1985), using exact eigenvectors and derivatives. It may be worthwhile to develop and extend this line of research to the case of Lanczos vectors.

(3) Implementation of Adaptive Strategy

The single biggest obstacle to the use of almost any nonlinear algorithm is that the analyst is often presented with a bewildering variety of error tolerances, step sizes, and solution parameters that must be chosen before the analysis can begin. One of the most important properties of a useful algorithm is that it insulates the analyst from as many of these decisions as possible. For instance, many integration schemes can be structured so that the algorithm (and not the user) determines the size of an appropriate integration step, adapting the step size to the data and solution of the problem without any intervention on the part of the analyst. The proposed reduced coordinate algorithm requires more parameters than a direct unreduced formulation, simply because it needs exactly the same data as the direct scheme, and also requires decisions as to the number of modes to be used and the error

tolerances for the algorithms to generate these modes (among other things). One of the most pressing requirements for the efficient implementation of this proposed method is that the algorithm be capable of finding reasonable values for these solution parameters, and of adaptively varying these quantities in response to the results obtained.

All three of these topics are presently being studied, and the algorithm is being modified in response to developments in these and other areas. The results of these tests and evaluations will be reported in appropriate journals.

References

Bathe, K.J., and S. Gracewski, 1981, On Nonlinear Dynamic Analysis Using Substructuring and Mode Superposition, Computers & Structures, Vol. 13, Pg. 699-707

Bayo, E.P., and E.L. Wilson, 1984a, Use of Ritz Vectors in Wave Propagation and Foundation Response, Earthquake Engineering and Structural Dynamics, Vol. 12, Pg. 499-505

Bayo, E.P., and E.L. Wilson, 1984b, Finite Element and Ritz Vector Techniques for the Solution to Three-Dimensional Soil-Structure Interaction Problems in the Time Domain, Engineering Computations, Vol. 1, No. 4, Dec. 1984, Pg. 298-311

Geschwinder, L.F. 1981, Nonlinear Dynamic Analysis by Modal Superposition, ASCE Journal of the Structural Division, Vol. 107, No. ST12, Dec. 1981, Pg. 2325-2336

Golub, G.H. and C.F. VanLoan, 1985, Matrix Computations, The Johns Hopkins University Press, Baltimore, MD.

Hughes, T.J.R, 1983, Analysis of Transient Algorithms with Particular Reference to Stability Behavior, in Computational Methods for Transient Analysis, T. Belytshcko and T.J.R. Hughes, Editors, Elsevier Science Publishers, North Holland

Idelsohn, S.R., and Cardona, A., 1985, A Reduction Method for Nonlinear Structural Dynamic Analysis, Computer Methods in Applied Mechanics and Engineering, No. 49, Pg. 253-279

Lanczos,C., 1950 An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, Journal of Research of the NBS, Vol. 45, No. 4, October 1950, Pg. 255-282

Mish, K.D., 1987, The Solution of Large Nonlinear Time-Dependent Problems Using Reduced Coordinates, Ph.D. Thesis, Department of Civil Engineering, The University of California at Davis

Newmark, N.M., 1959, A Method of Computation for Structural Dynamics, ASCE Journal of the Engineering Mechanics Division, Vol. 85, No. EM 3, July, 1959, Pg. 67-94

Noble, B., and J.W. Daniel, 1977, Applied Linear Algebra, Prentice-Hall

Nour-Omid, B., B.N. Parlett, and R.L. Taylor, 1983, A Newton-Lanczos method for Solution of Non-linear Finite Element Equations, Computers & Structures, Vol 16, No. 1-4, Pg. 241-252

Parlett, B.N., 1980a, A New Look at the Lanczos Algorithm, Linear Algebra and Its Applications, Vol. 29, Pg. 323-346

Parlett, B.N., 1980b, The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, N.J.

Stakgold, I., 1979, Green's Functions and Boundary Value Problems, John Wiley & Sons

Wilson, E.L., M. Yuan, and J.M. Dickens, 1982, Dynamic Analysis by Direct Superposition of Ritz Vectors, Earthquake Engineering and Structural Dynamics, Vol. 10, Pg. 813-821

Wilson, E.L., and Bayo, E.P., 1986, Use of Special Ritz Vectors in Dynamic Substructure Analysis, ASCE Journal of Structural Engineering, Vol. 112, No. 8, August 1986

DISTRIBUTION LIST

DTIC Alexandria, VA
GIDEP OIC, Corona, CA
NAVFACENGCOM Code 03, Alexandria, VA
NAVFACENGCOM - CHES DIV, Code FPO-1PL, Washington, DC
NAVFACENGCOM - LANT DIV, Library, Norfolk, VA
NAVFACENGCOM - NORTH DIV, Code 04AL, Philadelphia, PA
NAVFACENGCOM - PAC DIV, Library, Pearl Harbor, HI
NAVFACENGCOM - SOUTH DIV, Library, Charleston, SC
NAVFACENGCOM - WEST DIV, Library (Code 04A2.2), San Bruno, CA
PWC Code 101 (Library), Oakland, CA; Code 123-C, San Diego, CA; Code 420, Great Lakes, IL; Library (Code 134), Pearl Harbor, HI; Library, Guam, Mariana Islands; Library, Norfolk, VA; Library, Pensacola, FL; Library, Yokosuka JA; Tech Library, Subic Bay, RP

DEPARTMENT OF THE NAVY

NAVAL CIVIL ENGINEERING LABORATORY
PORT HUENEME, CALIFORNIA 93043 5003

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
DEPARTMENT OF THE NAVY
DOD-316

END

8-87

DTIC